Confidentiality level: external use

UM32Mx56 User Manual

V1.0.1

Unicmicro (Guangzhou) Co., Ltd.

http://www.unicmicro.com/

Copyright & Disclaimer

Contents

1 Documentation Conventions 2
1.1 List of Abbreviations for Registers 2
1.2 Glossary 2
2 Product Introduction 4
2.1 Overview 4
2.2 Main Features 5
3 Memory and Bus Architecture 8
3.1 System Architecture 8
3.2 Bus Architecture Diagram 9
3.3 Memory Mapping 10
4 Processor 12
4.1 Overview 12
4.2 Main Features 12
4.3 Functional Block Diagram 13
4.4 Core Register Set 13
5 System Configuration Utility (SCU) 14
5.1 Clock Block Diagram 14
5.2 Clock Selection 14
5.3 Reset Source 15
5.3.1 Internal POR 15
5.3.2 LVR 15
5.3.3 RESETEN 16
5.3.4 LOCKUP Reset 16
5.3.5 LVD Reset 16
5.3.6 WDT Reset 16
5.3.7 WWDT Reset 16
5.3.8 SOFT_RESETN 16
5.3.9 Module Reset 17
5.4 Low-power Mode 17
5.4.1 Sleep Mode 19
5.4.2 DeepSleep Mode 19
5.4.3 Stop Mode 20
5.5 System Register 22
5.5.1 System Control Register 0 SCU_CTRL0 (Offset: 000H) 23

5.5.2 System Control Protection Register SCU_CTRL_PROT (Offset: 008H) 24
5.5.3 Clock Control Register SCU_OSC_CTRL (Offset: 0x00CH) 25
5.5.4 Peripheral Module Clock Register SCU_PERI_CLKEN (Offset: 010H) 25
5.5.5 Reset Flag Register SCU_RESET_FLAG (Offset: 020H) 26
5.5.6 Peripheral Module Reset Control Register SCU_PERI_RESET (Offset: 024H) . 28
5.5.7 External Reset Filter Control Register SCU_EXTRST_CTRL (Offset: 028H) 30
5.5.8 Port PA Function Configuration Register SCU_PA_SEL (Offset: 030H) 30
5.5.9 Port PB Function Configuration Register SCU_PB_SEL (Offset: 034H) 33
5.5.10 Port PC Function Configuration Register SCU_PC_SEL (Offset: 038H) 36
5.5.11 Port PD Function Configuration Register SCU_PD_SEL (Offset: 03CH) 39
5.5.12 Port Analog-digital Configuration Register SCU_PAD_ADS (Offset: 054H) 41
5.5.13 Port Drive Capability Configuration Register SCU_PAD_DR (Offset: 060H) 43
5.5.14 Port Pull-up Configuration Register SCU_PAD_PU (Offset: 06CH) 45
5.5.15 Port Pull-down Configuration Register SCU_PAD_PD (Offset: 078H) 47
5.5.16 Port Open-drain Output Configuration Register SCU_PAD_OD (Offset: 084H) 48
5.5.17 Port Input Type Configuration Register SCU_PAD_CS (Offset: 090H) 50
5.5.18 Port Input Enable Register SCU_PAD_IE (Offset: 09CH) 52
5.5.19 Port Input Level Register SCU_PAD_STATUS (Offset: 0A4H) 53
5.5.20 Port Speed Configuration Register SCU_PAD_SR (Offset: 0A8H) 55
5.5.21 IO Control Protection Register SCU_IOCTRL_PROT (Offset: 0B4H) 57
5.5.22 LVD Configuration Register SCU_LVD_CFG (Offset: 0B8H) 57
5.5.23 PGA Control Register SCU_PGA_CFG (Offset: 0BCH) 59
5.5.24 External Reset Port Selection Register SCU_EXTRST_SEL (Offset: 0D0H) 61
5.5.25 Stop Mode Selection Register SCU_STOP_SEL (Offset: 0D4H) 61
5.5.26 Software Reset Register SCU_SOFT_RSTN (Offset: 0D8H) 62
5.5.27 Interrupt Vector Address Remapping Register SCU_VET_OFFSET (Offset: 0DCH) 62
5.5.28 LVD Interrupt Register SCU_LVD_INTR (Offset: FCH) 62
5.5.29 VREF Control Register SCU_VREF (Offset: 188H) 63

6 EFC 64
6.1 Overview 64
6.2 Main Features 64
6.3 eFlash Data Protection Mechanisms 64
6.3.1 eFlash Storage Area 65
6.3.2 EEPROM Configuration Words 66
6.3.3 SWD Hardware Disable 67
6.4 Register Description 68

6.4.1 EFC Control Register (EFC_CTRL) (Offset: 00H) 68
6.4.2 EFC OTP Status Register EFC_OTPSTATUS (Offset: 04H) 69
6.4.3 EFC KEY Input Register (EFC_KEY) (Offset: 08H) 71
6.4.4 EFC Interrupt Status Register (EFC_INTSTATUS) (Offset: 0CH) 71
6.4.5 EFC Interrupt Enable Register EFC_INTEN (Offset: 10H) 73
6.4.6 EFC Erase/Write Control Register (EFC_EP_CTRL) (Offset: 14H) 74
6.4.7 EFC LDO TRIM Register (EFC_LDOTRIM) (Offset: 1CH) 75
6.4.8 EFC RCH TRIM Register (EFC_RCHTRIM) (Offset: 20H) 75
6.4.9 EFC RCL TRIM Register (EFC_RCLTRIM) (Offset: 24H) 75
6.4.10 EFC PGA0 TRIM Register (EFC_PGA0TRIM) (Offset: 28H) 75
6.4.11 EFC PGA1 TRIM Register (EFC_PGA1TRIM) (Offset: 2CH) 76
6.4.12 EFC ADC TRIM Register (EFC_ADCTRIM) (Offset: 30H) 76
6.5 Software Process 76
6.5.1 Write Operation 76
6.5.2 Erase Operation 78
7 NVIC 79
7.1 Overview 79
7.2 Main Features 79
7.3 Interrupt Source 80
8 GPIO 81
8.1 Overview 81
8.2 Main Features 81
8.3 Register Description 81
8.3.1 GPIO Data Direction Register (GPIO_DIR) (Offset: 00H) 82
8.3.2 GPIO Output Set Register (GPIO_SET) (Offset: 08H) 82
8.3.3 GPIO Output Clear Register (GPIO_CLR) (Offset: 0CH) 83
8.3.4 GPIO Output Pin Mapping Register (GPIO_ODATA) (Offset: 10H) 83
8.3.5 GPIO Input Pin Mapping Register (GPIO_IDATA) (Offset: 14H) 83
8.3.6 GPIO Interrupt Enable Register (GPIO_IEN) (Offset: 18H) 84
8.3.7 GPIO Interrupt Trigger Mode Register (GPIO_IS) (Offset: 1CH) 84
8.3.8 GPIO Interrupt Edge Trigger Setting Register (GPIO_IBE) (Offset: 20H) 84
8.3.9 GPIO Interrupt High/Low Level Trigger Setting Register (GPIO_IEV) (Offset: 24H) 85
8.3.10 GPIO Interrupt Status Clear Register (GPIO_IC) (Offset: 28H) 85
8.3.11 GPIO Raw Interrupt Status Register (GPIO_RIS) (Offset: 2CH) 85
8.3.12 GPIO Mask Interrupt Status Register (GPIO_MIS) (Offset: 30H) 86
8.3.13 GPIO Input Filter Selection Register (GPIO_FLS) (Offset: 34H) 86

8.4 Operation Procedure....86
8.4.1 IO....86
8.4.2 Interrupt Trigger Mode....87
8.4.3 Clearing Interrupt....87
8.4.4 Input Filtering....87
9 UART0/1....88
9.1 Overview....88
9.2 Main Features....88
9.3 Register Description....88
9.3.1 Interrupt Status Register UART_ISR (Offset: 00H)....89
9.3.2 Interrupt Enable Register UART_IER (Offset: 04H)....89
9.3.3 Control Register UART_CR (Offset: 08H)....90
9.3.4 Transmit Data Register UART_TDR (Offset: 0CH)....91
9.3.5 Receive Data Register UART_RDR (Offset: 10H)....91
9.3.6 Baud Rate Parameter Low Register UART_BRR (Offset: 14H)....92
9.3.7 Delay Timeout Register UART_DLTOR (Offset: 18H)....92
9.4 Operation Procedure....92
9.4.1 Transmission and Reception via UART....92
9.4.2 UART Initialization....93
9.4.3 UART Transmitting Byte....93
9.4.4 UART Receiving Byte....93
10 I2C....94
10.1 Overview....94
10.2 Main Features....94
10.3 Register Description....94
10.3.1 I2C Master Configuration Register I2C_MCFG (Offset: 00H)....95
10.3.2 I2C Master Control Register I2C_MCON (Offset: 04H)....96
10.3.3 I2C Master Interrupt Enable Register I2C_MIE (Offset: 08H)....96
10.3.4 I2C Master Interrupt Flag Register I2C_MIF (Offset: 0CH)....97
10.3.5 I2C Master Status Register I2C_MSTA (Offset: 10H)....98
10.3.6 I2C Master Baud Rate Register I2C_MBRG (Offset: 14H)....99
10.3.7 I2C Master Transmit and Receive Buffer Register I2C_MBUF (Offset: 18H)....99
10.3.8 I2C Master Timing Control Register I2C_MTIM (Offset: 1CH)....99
10.3.9 I2C Master Timeout Register I2C_MTO (Offset: 20H)....100
10.3.10 I2C Slave Control Register I2C_SCON (Offset: 24H)....100
10.3.11 I2C Slave Interrupt Enable Register I2C_SIE (Offset: 28H)....101
10.3.12 I2C Slave Interrupt Status Register I2C_SIF (Offset: 2CH)....102

10.3.13 I2C Slave Status Register I2C_SSTA (Offset: 30H) 104
10.3.14 I2C Slave Transmit and Receive Buffer Register I2C_SBUF (Offset: 34H) 105
10.3.15 I2C Slave Address Register 0 I2C_SADR0 (Offset: 38H) 105
10.3.16 I2C Slave Address Register 1 I2C_SADR1 (Offset: 3CH) 105
10.3.17 I2C Slave Address Register 2 I2C_SADR2 (Offset: 40H) 105
10.3.18 I2C Slave Address Register 3 I2C_SADR3 (Offset: 44H) 105
10.4 Operation Procedure 106
10.4.1 Master Transmit 106
10.4.2 Master Receive 107
10.4.3 Slave Transmit 108
10.4.4 Slave Receive 109
11 SPI 110
11.1 Overview 110
11.2 Main Features 110
11.3 Register Description 111
11.3.1 SPI Configuration Register (SPI_CR) (Offset: 00H) 111
11.3.2 SPI Master Mode Control Register 0 (SPI_CS0) (Offset: 04H) 113
11.3.3 SPI Master Mode Control Register 1 (SPI_CS1) (Offset: 08H) 115
11.3.4 SPI Process Control Register (SPI_OPCR) (Offset: 14H) 116
11.3.5 SPI Interrupt Control Register (SPI_IE) (Offset: 18H) 117
11.3.6 SPI Interrupt Flag Register (SPI_IF) (Offset: 1CH) 118
11.3.7 SPI Transmit Buffer Register (SPI_TXBUF) (Offset: 20H) 119
11.3.8 SPI Receive Buffer Register (SPI_RXBUF) (Offset: 24H) 119
11.3.9 SPI DMA Receive Setting Register (SPI_DMARXLEV) (Offset: 28H) 120
11.3.10 SPI DMA Transmit Setting Register (SPI_DMATXLEV) (Offset: 2CH) 120
11.4 Operation Procedure 120
11.4.1 Initialization Process 121
11.4.2 Transmission Process 121
11.4.3 Reception Process 122
11.4.4 SPI DMA Transmission Process 122
11.4.5 SPI DMA Reception Process 123
12 ATIMER 125
12.1 Overview 125
12.2 Main Features 125
12.3 ATIMER Block Diagram 126
12.4 Functional Description 127
12.4.1 Timing Unit 127

12.4.2 Counter Operation Mode 129
12.4.3 Up-counting Mode 129
12.4.4 Down-counting Mode 132
12.4.5 Center-aligned Counting Mode 135
12.4.6 Repetition Counter 137
12.4.7 Preload Register 138
12.4.8 Counter Operating Clock 139
12.4.9 Internal Clock Mode 139
12.4.10 External Clock Mode 1 139
12.4.11 External Clock Mode 2 142
12.4.12 Internal Trigger Signal (ITRx) 144
12.4.13 Capture/Compare Channels 144
12.4.14 Input Capture Mode 146
12.4.15 Software Force Output 148
12.4.16 Output Compare Mode 148
12.4.17 PWM Output 149
12.4.18 Asymmetric PWM Phase-Shift Control 151
12.4.19 Complementary Output and Dead-time Insertion 152
12.4.20 Break Function 153
12.4.21 6-step PWM Output 155
12.4.22 Single-pulse Output 156
12.4.23 External Event Clearing OCxREF 158
12.4.24 Encoder Interface Mode 159
12.4.25 ATIMER Slave Mode 161
12.4.26 Reset Mode 161
12.4.27 Gated Mode 162
12.4.28 Trigger Mode 163
12.4.29 External Clock Counting Mode Triggered by External Events 164
12.4.30 DMA Access 165
12.4.31 Input XOR Function 166
12.4.32 Debug Mode 166
12.4.33 DMA Burst 166
12.5 Register Description 167
12.5.1 ATIMER Control Register 1 ATIMER_CR1 (Offset: 00H) 168
12.5.2 ATIMER Control Register 2 ATIMER_CR2 (Offset: 04H) 171
12.5.3 ATIMER Slave Mode Control Register ATIMER_SMCR (Offset: 08H) 173
12.5.4 ATIMER DMA and Interrupt Enable Register ATIMER_DIER (Offset: 0CH) 175
12.5.5 ATIMER Status Register ATIMER_SR (Offset: 10H) 177

12.5.6 ATIMER Event Generation Register ATIMER_EGR (Offset: 14H) 179
12.5.7 ATIMER Capture/Compare Mode Register 1 ATIMER_CCMR1 (Offset: 18H) 180
12.5.8 ATIMER Capture/Compare Mode Register 2 ATIMER_CCMR2 (Offset: 1CH) 183
12.5.9 ATIMER Capture/Compare Enable Register ATIMER_CCER (Offset: 20H) 187
12.5.10 ATIMER Counter Register ATIMER_CNT (Offset: 24H) 189
12.5.11 ATIMER Prescaler Register ATIME_PSC (Offset: 28H) 190
12.5.12 ATIMER Auto-reload Register ATIMER_ARR (Offset: 2CH) 190
12.5.13 ATIMER Repetition Counter Register ATIMER_RCR (Offset: 30H) 190
12.5.14 ATIMER Capture/Compare Register 1 ATIMER_CCR1 (Offset: 34H) 191
12.5.15 ATIMER Capture/Compare Register 2 ATIMER_CCR2 (Offset: 38H) 191
12.5.16 ATIMER Capture/Compare Register 3 ATIMER_CCR3 (Offset: 3CH) 192
12.5.17 ATIMER Capture/Compare Register 4 ATIMER_CCR4 (Offset: 40H) 192
12.5.18 ATIMER Break and Dead-time Control Register ATIMER_BDTR (Offset: 44H) .. 193
12.5.19 ATIMER DMA Control Register ATIMER_DCR (Offset: 48H) 195
12.5.20 ATIMER DMA Access Register ATIMER_DMAR (Offset: 4CH) 195
12.5.21 ATIMER Capture/Compare Mode Register 3 ATIMER_CCMR3 (Offset: 50H) 196
12.5.22 ATIMER Capture/Compare Register 5 ATIMER_CCR5 (Offset: 54H) 197
12.5.23 ATIMER PWM Phase-shift Enable Register ATIMER_PMEN (Offset: 58H) .. 198
12.5.24 ATIMER Phase-shift Offset Register 1 ATIMER_PMC1 (Offset: 5CH) 198
12.5.25 ATIMER Phase-shift Offset Register 2 ATIMER_PMC2 (Offset: 60H) 198
12.5.26 ATIMER Phase-shift Offset Register 3 ATIMER_PMC3 (Offset: 64H) 199
12.5.27 ATIMER Break Input Control Register ATIMER_BKCTL (Offset: 68H) 199
12.6 Operation Procedure 200
12.6.1 Timer Counting Mode 200
12.6.2 PWM Mode 201
12.6.3 Input Capture Mode 202
12.6.4 Complementary Output and Dead-time Insertion 203
12.6.5 Break Function 203
12.6.6 Encoder Interface Mode 204
12.6.7 DMA Mode 205

13 General-purpose Timer GTIMER0 206
13.1 Overview 206
13.2 Main Features 206
13.3 GTIMER0 Block Diagram 207

13.4 Functional Description 207
13.4.1 Timing Unit 207
13.4.2 Timer Operating Modes 208
13.4.2.1 Up-counting 209
13.4.2.2 Down-counting 210
13.4.2.3 Center-aligned Counting 211
13.4.3 Counter Operating Clock 212
13.4.3.1 APBCLK 212
13.4.3.2 GTIMER_CHx (x = 1, 2) Pin Input: External Clock Mode 1 213
13.4.3.3 GTIMER_ETR Pin Input: External Clock Mode 2 213
13.4.4 Internal Trigger Signal (ITRx) 214
13.4.5 Capture/Compare Channels 214
13.4.5.1 Capture Channel 214
13.4.5.2 Compare Channel 215
13.4.6 Input Capture Mode 216
13.4.7 PWM Input Mode 217
13.4.8 Software-forced Output 218
13.4.9 Output Compare Mode 218
13.4.10 PWM Output 220
13.4.10.1 PWM Edge-aligned Mode 220
13.4.10.2 PWM Center-aligned Mode 221
13.4.11 Output Compare Fast Enable Mode 221
13.4.12 Single-pulse Output 222
13.4.13 Clearing OCxREF via External Event 224
13.4.14 Encoder Interface Mode 225
13.4.15 GTIMER Slave Mode 226
13.4.15.1 Reset Mode 227
13.4.15.2 Gated Mode 228
13.4.15.3 Trigger Mode 229
13.4.16 Timer Synchronization 229
13.4.17 DMA Burst 230
13.4.18 Input XOR Function 230
13.4.19 Debug Mode 231
13.5 Register Description 231
13.5.1 GTIMER Registers 231
13.5.2 Control Register 1 GTIMER_CR1 (Offset: 00H) 232
13.5.3 Control Register 2 GTIMER_CR2 (Offset: 04H) 233
13.5.4 Slave Mode Control Register GTIMER_SMCR (Offset: 08H) 235

13.5.5 DMA and Interrupt Enable Register GTIMER_DIER (Offset: 0CH) 237
13.5.6 Status Register GTIMER_SR (Offset: 10H) 239
13.5.7 Event Generation Register GTIMER_EGR (Offset: 14H) 240
13.5.8 Capture/Compare Mode Register 1 GTIMER_CCMR1 (Offset: 18H) 241
13.5.9 Capture/Compare Mode Register 2 GTIMER_CCMR2 (Offset: 1CH) 245
13.5.10 Capture/Compare Enable Register GTIMER_CCER (Offset: 20H) 249
13.5.11 Counter Register GTIMER_CNT (Offset: 24H) 250
13.5.12 Prescaler Register GTIMER_PSC (Offset: 28H) 250
13.5.13 Auto-reload Register GTIMER_ARR (Offset: 2CH) 250
13.5.14 Capture/Compare Register 1 GTIMER_CCR1 (Offset: 34H) 250
13.5.15 Capture/Compare Register 2 GTIMER_CCR2 (Offset: 38H) 251
13.5.16 Capture/Compare Register 3 GTIMER_CCR3 (Offset: 3CH) 251
13.5.17 Capture/Compare Register 4 GTIMER_CCR4 (Offset: 40H) 251
13.5.18 DMA Control Register GTIMER_DCR (Offset: 48H) 252
13.5.19 DMA Access Register GTIMER_DMAR (Offset: 4CH) 253

14 General-purpose Timers GTIMER1/2 254
14.1 Overview 254
14.2 Main Features 254
14.3 GTIMER1/2 Block Diagram 255
14.4 Functional Description 255
14.4.1 Timing Unit 255
14.4.2 Timer Operating Modes 256
14.4.2.1 Up-counting Mode 257
14.4.2.2 Down-counting Mode 258
14.4.2.3 Center-aligned Counting Mode 259
14.4.3 Counter Clock 260
14.4.3.1 APBCLK 260
14.4.3.2 GTIMER_CHx (x = 1, 2) Pin Input: External Clock Mode 1 261
14.4.3.3 GTIMER_ETR Pin Input: External Clock Mode 2 262
14.4.4 Internal Trigger Signal (ITRx) 262
14.4.5 Capture/Compare Channels 262
14.4.5.1 Capture Channel 263
14.4.5.2 Compare Channel 263
14.4.6 Input Capture Mode 264
14.4.7 PWM Input Mode 265
14.4.8 Software-forced Output 266
14.4.9 Output Compare Mode 266
14.4.10 PWM Output 268

14.4.10.1 PWM Edge-aligned Mode 268
14.4.10.2 PWM Center-aligned Mode 269
14.4.11 Output Compare Fast Enable Mode 269
14.4.12 Single-pulse Output Mode 270
14.4.13 Clearing OCxREF via External Event 272
14.4.14 Encoder Interface Mode 273
14.4.15 GTIMER Slave Mode 274
14.4.15.1 Reset Mode 275
14.4.15.2 Gated Mode 276
14.4.15.3 Trigger Mode 277
14.4.16 Timer Synchronization 277
14.4.17 DMA Burst 278
14.4.18 Input XOR Function 278
14.4.19 Debug Mode 279
14.5 Register Description 279
14.5.1 GTIMER Registers 279
14.5.2 Control Register 1 GTIMER_CR1 (Offset: 00H) 280
14.5.3 Control Register 2 GTIMER_CR2 (Offset: 04H) 281
14.5.4 Slave Mode Control Register GTIMER_SMCR (Offset: 08H) 282
14.5.5 DMA and Interrupt Enable Register GTIMER_DIER (Offset: 0CH) 284
14.5.6 Status Register GTIMER_SR (Offset: 10H) 287
14.5.7 Event Generation Register GTIMER_EGR (Offset: 14H) 288
14.5.8 Capture/Compare Mode Register 1 GTIMER_CCMR1 (Offset: 18H) 289
14.5.9 Capture/Compare Mode Register 2 GTIMER_CCMR2 (Offset: 1CH) 292
14.5.10 Capture/Compare Enable Register GTIMER_CCER (Offset: 20H) 296
14.5.11 Counter Register GTIMER_CNT (Offset: 24H) 297
14.5.12 Prescaler Register GTIMER_PSC (Offset: 28H) 297
14.5.13 Auto-reload Register GTIMER_ARR (Offset: 2CH) 297
14.5.14 Capture/Compare Register 1 GTIMER_CCR1 (Offset: 34H) 298
14.5.15 Capture/Compare Register 2 GTIMER_CCR2 (Offset: 38H) 298
14.5.16 Capture/Compare Register 3 GTIMER_CCR3 (Offset: 3CH) 298
14.5.17 Capture/Compare Register 4 GTIMER_CCR4 (Offset: 40H) 299
14.5.18 DMA Control Register GTIMER_DCR (Offset: 48H) 299
14.5.19 DMA Access Register GTIMER_DMAR (Offset: 4CH) 300

15 LPTIMER 301
15.1 Overview 301
15.2 Main Features 301
15.3 Block Diagram 302

15.4 Timer Function 302
15.4.1 Single-shot Counting and Continuous Counting 302
15.4.2 ETR Trigger Counting 303
15.4.3 ETR Pulse Counting 303
15.4.4 Timeout Mode 303
15.5 Capture/Compare Function 304
15.5.1 PWM Output 304
15.5.2 Input Capture 304
15.5.3 Debug Mode 304
15.6 Register Description 304
15.6.1 LPTIMER Control Register 1 LPTIMER_CR1 (Offset: 00H) 305
15.6.2 LPTIMER Control Register 2 LPTIMER_CR2 (Offset: 04H) 305
15.6.3 LPTIMER Interrupt Enable Register LPTIMER_IER (Offset: 08H) 307
15.6.4 LPTIMER Interrupt Flag Register LPTIMER_SR (Offset: 0CH) 308
15.6.5 LPTIMER Counter Value Register LPTIMER_CNT (Offset: 10H) 309
15.6.6 LPTIMER Capture/Compare Configuration Register 1 LPTIMER_CCMCFG1 (Offset: 14H) 309
15.6.7 LPTIMER Capture/Compare Configuration Register 2 LPTIMER_CCMCFG2 (Offset: 18H) 310
15.6.8 LPTIMER Auto-reload Register LPTIMER_ARR (Offset: 1CH) 310
15.6.9 LPTIMER Capture/Compare Register 1 LPTIMER_CCR1 (Offset: 20H) 311
15.6.10 LPTIMER Capture/Compare Register 2 LPTIMER_CCR2 (Offset: 24H) 311
15.6.11 LPTIMER Counter Value Load Register LPTIMER_LOAD (Offset: 28H) 311
15.6.12 LPTIMER Counter Buffer Register LPTIMER_BUFFER (Offset: 2CH) 311
15.7 Operation Procedure 312
15.7.1 General-purpose Timer 312
15.7.2 Input Capture Function with DMA 312
15.7.3 PWM Output 313
15.7.4 ETR Pulse Trigger Counting Mode 313
15.7.5 ETR Pulse Counting Mode 314
15.7.6 Timeout Mode 315
16 DMA 316
16.1 Overview 316
16.2 Main Features 316
16.3 Register Description 316
16.3.1 DMA Channel Source Transfer Address Register (DMA_SRCADDR_Cx) (Offset: 20 * x + 00H) (x = 0, 1) 317
16.3.2 DMA Channel Destination Transfer Address Register (DMA_DSTADDR_Cx)

(Offset: 20 * x + 04H) (x = 0, 1) 317
16.3.3 DMA Channel Control Information Register (DMA_CHCTRL_Cx) (Offset: 20 * x + 08H) (x = 0, 1) 318
16.3.4 DMA Channel Transfer Status Register (DMA_CHSTS_Cx) (Offset: 20 * x + 0CH) (x = 0, 1) 320
16.3.5 DMA Channel Source Peripheral Selection Register (DMA_CHSPER_Cx) (Offset: 20 * x + 10H) (x = 0, 1) 320
16.3.6 DMA Channel Destination Peripheral Selection Register (DMA_CHDPER_Cx) (Offset: 20 * x + 14H) (x = 0, 1) 321
16.3.7 DMA Controller Enable Register (DMA_EN) (Offset: 40H) 323
16.3.8 DMA Soft Reset Register (DMA_SOFT_RESET) (Offset: 44H) 323
16.3.9 DMA Interrupt Status Register (DMA_INT_STATUS) (Offset: 48H) 323
16.3.10 DMA Interrupt Mask Register (DMA_INT_MASK) (Offset: 4CH) 324
16.3.11 DMA Peripheral Request Register (DMA_PER_REQ) (Offset: 54H) 324
16.4 Operation Procedure 325

17 CRC 327
17.1 Overview 327
17.2 Register Description 327
17.2.1 Data Register CRC_DATA (Offset: 00H) 327
17.2.2 Initial Value Register CRC_INIT (Offset: 04H) 328
17.2.3 Control Register CRC_CTRL (Offset: 08H) 328
17.3 Operation Procedure 328

18 RNG 330
18.1 Overview 330
18.2 Main Features 330
18.3 Register Description 330
18.3.1 Random Number Control Register RNG_CR (Offset: 0E0H) 330
18.3.2 Random Number Seed Register RNG_SEED (Offset: 0E4H) 331
18.3.3 Random Number Data Register (Offset: 0E8H) 331
18.4 Operation Procedure 331

19 WDT 332
19.1 Overview 332
19.2 Main Features 332
19.3 Register Description 332
19.3.1 Load Register WDT_LOAD (Offset: 00H) 333
19.3.2 Counter Register WDT_CNT (Offset: 04H) 333
19.3.3 Control Register WDT_CTRL (Offset: 08H) 333

19.3.4 Clear Register WDT_CLR (Offset: 0CH) 334
19.3.5 RAW Interrupt Status Register WDT_INTRAW (Offset: 10H) 334
19.3.6 Mask Interrupt Status Register WDT_MINTS (Offset: 14H) 334
19.3.7 STALL Control Register WDT_STALL (Offset: 18H) 334
19.3.8 LOCK Register WDT_LOCK (Offset: 1CH) 335
19.4 Operation Procedure 335
19.4.1 WDT Timer Configuration 335
19.4.2 WDT Watchdog Feeding Configuration 336
20 WWDT 337
20.1 Overview 337
20.2 Main Features 337
20.3 Register Description 337
20.3.1 Control Register WWDT_CON (Offset: 00H) 338
20.3.2 Configuration Register WWDT_CFG (Offset: 04H) 338
20.3.3 Counter Register WWDT_CNT (Offset: 08H) 339
20.3.4 Interrupt Enable Register WWDT_IE (Offset: 0CH) 339
20.3.5 Interrupt Flag Register WWDT_IF (Offset: 10H) 339
20.4 Operation Procedure 340
21 ADC 341
21.1 Overview 341
21.2 Main Features 341
21.3 ADC Pin Description 342
21.4 Register Description 343
21.4.1 ADC Interrupt and Status Register (ADC_ISR) (Offset: 000H) 344
21.4.2 ADC Interrupt Enable Register (ADC_IER) (Offset: 004H) 346
21.4.3 ADC Control Register (ADC_CR) (Offset: 008H) 347
21.4.4 ADC Configuration Register (ADC_CFGR) (Offset: 00CH) 347
21.4.5 ADC Sampling Time Control Register (ADC_SMTR) (Offset: 010H) 350
21.4.6 ADC Channel Control Register (ADC_CHER) (Offset: 014H) 350
21.4.7 ADC Channel Selection Register (ADC_SEQ) (Offset: 018H) 351
21.4.8 ADC Analog Watchdog Threshold Register (ADC_HLTR) (Offset: 020H) 352
21.4.9 ADC Injected Channel Control Register (ADC_IJCHER) (Offset: 024H) 352
21.4.10 ADC Injected Channel Selection Register (ADC_IJSEQ) (Offset: 028H) 353
21.4.11 ADC Data Register 0 (ADC_DR0) (Offset: 030H) 354
21.4.12 ADC Data Register 1 (ADC_DR1) (Offset: 034H) 354
21.4.13 ADC Data Register 2 (ADC_DR2) (Offset: 038H) 354
21.4.14 ADC Data Register 3 (ADC_DR3) (Offset: 03CH) 354

21.4.15 ADC Data Register 4 (ADC_DR4) (Offset: 040H) 354
21.4.16 ADC Data Register 5 (ADC_DR5) (Offset: 044H) 354
21.4.17 ADC Data Register 6 (ADC_DR6) (Offset: 048H) 355
21.4.18 ADC Data Register 7 (ADC_DR7) (Offset: 04CH) 355
21.4.19 ADC Injected Channel Data Register 0 (ADC_IJDR0) (Offset: 050H) 355
21.4.20 ADC Injected Channel Data Register 1 (ADC_IJDR1) (Offset: 054H) 355
21.4.21 ADC Injected Channel Data Register 2 (ADC_IJDR2) (Offset: 058H) 355
21.4.22 ADC Injected Channel Data Register 3 (ADC_IJDR3) (Offset: 05CH) 356
21.4.23 ADC Oversampling Average Result Register (ADC_DR_OVS) (Offset: 060H) 356
21.4.24 ADC Hardware Trigger Enable Register (ADC_HDT) (Offset: 0A8H) 356
21.4.25 ADC Injected Channel Hardware Trigger Enable Register (ADC_IJHDT) (Offset: 0ACH) 358
21.4.26 ADC Channel Sampling Time Configuration Register 0 (ADC_SMTR0) (Offset: 0B0H) 361
21.4.27 ADC Channel Sampling Time Configuration Register 1 (ADC_SMTR1) (Offset: 0B4H) 362
21.4.28 ADC Injected Channel Sampling Time Configuration Register (ADC_IJSMTR) (Offset: 0B8H) 363
21.5 ADC Operation Procedure 365
21.5.1 Single-channel A/D Conversion in Single-shot Scan Mode 365
21.5.2 Multi-channel A/D Conversion in Single-shot Scan Mode 366
21.5.3 Single-channel A/D Conversion in Continuous Scan Mode 367
21.5.4 Multi-channel A/D Conversion in Continuous Scan Mode 368
21.5.5 A/D Conversion Initiated by Hardware-triggered Event 369
21.5.6 A/D Conversion in Normal Injection Mode of Injected Channels 370
21.5.7 A/D Conversion in Automatic Injection Mode of Injected Channels 372
21.5.8 Notes 374
21.6 Workflow of ADC Sampling via PGA Buffer 375
21.6.1 Diagram of ADC Sampling via PGA Buffer 375
21.6.2 Flowchart of ADC Sampling via PGA Buffer 375
21.6.3 Workflow of ADC Sampling via PGA Buffer 375

22 COMP 377
22.1 Overview 377
22.2 Main Features 377
22.3 COMP Functional Block Diagram 378
22.4 COMP Filtering Modes 379
22.5 Window Function 380

22.6 Register Description 380
22.6.1 COMP0 Control Register COMP0_CFG (Offset: 118H) 381
22.6.2 COMP1 Control Register COMP1_CFG (Offset: 11CH) 382
22.6.3 COMP0 Polling Register COMP0_POLL (Offset: 128H) 384
22.6.4 COMP1 Polling Register COMP1_POLL (Offset: 12CH) 385
22.6.5 COMP0 Polling MASK Register COMP0_POLLMASK (Offset: 138H) 386
22.6.6 COMP1 Polling MASK Register COMP1_POLLMASK (Offset: 13CH) 386
22.6.7 COMP0 Filter Configuration Register COMP0_LVSET (Offset: 148h) 386
22.6.8 COMP1 Filter Configuration Register COMP1_LVSET (Offset: 14CH) 387
22.6.9 COMP0 WINDOW Configuration Register COMP0_WINCFG (Offset: 158H) 387
22.6.10 COMP1 WINDOW Configuration Register COMP1_WINCFG (Offset: 15CH) 390
22.6.11 COMP0 Status Register COMP0_INTSTAT (Offset: 168H) 394
22.6.12 COMP1 Status Register COMP1_INTSTAT (Offset: 16CH) 395
22.6.13 COMP0 Interrupt Enable Register COMP0_INTEN (Offset: 178H) 396
22.6.14 COMP1 Interrupt Enable Register COMP1_INTEN (Offset: 17CH) 396
22.7 Operation Procedure 397
23 DIV 398
23.1 Overview 398
23.2 Main Features 398
23.3 Register Description 398
23.3.1 Dividend Register DIV_DIVIDEND (Offset: 00H) 399
23.3.2 Divisor Register DIV_DIVISOR (Offset: 04H) 399
23.3.3 Remainder Register DIV_REMAIN (Offset: 08H) 399
23.3.4 Quotient Register DIV_QUITIENT (Offset: 0CH) 399
23.3.5 Status Register DIV_STATUS (Offset: 10H) 399
23.4 Operation Procedure 400
24 SQRT 401
24.1 Overview 401
24.2 Main Features 401
24.3 Register Description 401
24.3.1 SQRT Radicand Register (SQRT_VALUE) (Offset: 00H) 401
24.3.2 SQRT Square Root Operation Result Register (SQRT_RESULT) (Offset: 04H) 402
24.3.3 SQRT Status Register (SQRT_SR) (Offset: 08H) 402
24.4 Operation Procedure 402

25 SysTick 403
25.1 Overview 403
25.2 Register Description 403
25.2.1 Control and Status Register SysTick_CTRL (Offset: 00H) 404
25.2.2 Reload Value Register SysTick_LOAD (Offset: 04H) 404
25.2.3 Current Value Register SysTick_VAL (Offset: 08H) 404
25.3 Operation Procedure 405
26 Debug Support (DBG) 406
27 Revision History 407

List of Figures

Figure 3-1: Bus Architecture Diagram 9
Figure 3-2: Memory Address Mapping Diagram 10
Figure 4-1: Cortex-M0+ Functional Block Diagram 13
Figure 4-2: Cortex-M0+ Register Set 13
Figure 5-1: Clock Module Block Diagram 14
Figure 6-1: eFlash Storage Area 66
Figure 6-2: EEPROM8 Keyword Configuration Diagram 66
Figure 6-3: Write Operation Process 77
Figure 6-4: Sector Erase Operation Process 78
Figure 6-5: Chip Erase Operation Process 78
Figure 10-1: Master Transmit Flowchart 106
Figure 10-2: Master Receive Flowchart 107
Figure 10-3: Slave Transmit Flowchart 108
Figure 10-4: Slave Receive Flowchart 109
Figure 12-1: ATIMER Block Diagram 126
Figure 12-2: Counter Timing Diagram with Prescaler Division Changing from 1 to 2 128
Figure 12-3: Counter Timing Diagram with Prescaler Division Changing from 1 to 4 128
Figure 12-4: Up-counting Waveform, Internal Clock not Divided 130
Figure 12-5: Up-counting Waveform, Internal Clock Divided by 2 130
Figure 12-6: Update Event when ARPE = 0 (ATIMER_ARR not Preloaded) 131
Figure 12-7: Update Event when ARPE = 1 (ATIMER_ARR Preloaded) 131
Figure 12-8: Down-counting Waveform, Internal Clock not Divided 133
Figure 12-9: Down-counting Waveform, Internal Clock Divided by 2 133
Figure 12-10: Update Event when ARPE = 0 (ATIMER_ARR not Preloaded) 134
Figure 12-11: Update Event when ARPE = 1 (ATIMER_ARR Preloaded) 134
Figure 12-12: Center-aligned Counter Timing Diagram, ATIMER_PSC = 0, ATIMER_ARR = 0x6 135
Figure 12-13: Counter Timing Diagram, Update Event When ARPE = 1 (Counter Underflow) 136
Figure 12-14: Counter Timing Diagram, Update Event When ARPE = 1 (Counter Overflow).. 136
Figure 12-15: Examples of Update Rates in Different Modes and the Setting of ATIMER_RCR Register 137
Figure 12-16: Internal Clock Source Mode, Clock Division Factor being 1 139
Figure 12-17: Example of External Clock Connection 140
Figure 12-18: Timing Diagram for External Clock Mode 1 140
Figure 12-19: Timing Diagram for External Clock Mode 1 141
Figure 12-20: Block Diagram of External Trigger Input 142

Figure 12-21: Timing Diagram 1 for External Clock Mode 2 142
Figure 12-22: Timing Diagram for External Clock Mode 2 143
Figure 12-23: Capture/Compare Channel (Input Section of Channel 1)..... 144
Figure 12-24: Main Circuit of Capture/Compare Channel 1 145
Figure 12-25: Output Section of Capture/Compare Channels (Channels 1−3)..... 145
Figure 12-26: Output Section of Capture/Compare Channel 4 146
Figure 12-27: PWM Input Capture Mode Timing Diagram..... 147
Figure 12-28: Output Compare Mode, Toggle OC1..... 149
Figure 12-29: Edge-aligned PWM Waveform (ARR = 7) 150
Figure 12-30: Center-aligned PWM Waveform (ARR = 7)..... 151
Figure 12-31: Phase-shifted PWM Waveform (ARR = 7)..... 152
Figure 12-32: Complementary Output with Dead-time Insertion 153
Figure 12-33: Dead-time Waveform with Delay Greater than Negative Pulse..... 153
Figure 12-34: Dead-time Waveform with Delay Greater than Positive Pulse..... 153
Figure 12-35: Output in Response to Break 155
Figure 12-36: Example of Generating Six-step PWM Using COM (OSSR = 1) 156
Figure 12-37: Example of Single-pulse Mode..... 157
Figure 12-38: ETR Signal Clearing OCxREF of ATIMER 158
Figure 12-39: Example of Counter Operation in Encoder Mode..... 160
Figure 12-40: Timing Diagram in Reset Mode 162
Figure 12-41: Timing Diagram in Gated Mode..... 163
Figure 12-42: Timing Diagram in Trigger Mode 163
Figure 12-43: Timing Diagram in External Clock Mode 2 and Trigger Mode..... 164
Figure 13-1: GTIMER0 Block Diagram 207
Figure 13-2: GTIMER Timing Diagram with Prescaler Division Changing from 1 to 2 208
Figure 13-3: Up-counting Waveform 209
Figure 13-4: Down-counting Waveform..... 210
Figure 13-5: Center-aligned Counter Timing Diagram..... 212
Figure 13-6: APBCLK Clock Source Mode with a Clock Prescaler Factor of 1..... 212
Figure 13-7: Capture/Compare Channel (Input Section of Channel 1)..... 215
Figure 13-8: Main Circuit of Capture/Compare Channel 1 215
Figure 13-9: Output Section of Capture/Compare Channel..... 215
Figure 13-10: PWM Input Capture..... 218
Figure 13-11: Output Compare Mode, Toggling OC1..... 219
Figure 13-12: Edge-aligned PWM Waveform (ARR = 7) 220
Figure 13-13: Center-aligned PWM Waveform (APR = 7) 221
Figure 13-14: Example of Single-pulse Mode..... 223
Figure 13-15: GTIMER_ETR Signal Clearing OCxREF of GTIMER..... 225

Figure 13-16: Example of Counter Operation in Encoder Mode 226
Figure 13-17: Timing Diagram in Reset Mode 227
Figure 13-18: Timing Diagram in Gated Mode 228
Figure 13-19: Timing Diagram in Trigger Mode 229
Figure 14-1: GTIMER Block Diagram 255
Figure 14-2: GTIMER Timing Diagram with Prescaler Division Changing from 1 to 2 256
Figure 14-3: Up-counting Waveform 257
Figure 14-4: Down-counting Waveform 258
Figure 14-5: Center-aligned Counter Timing Diagram 260
Figure 14-6: APBCLK Clock Source Mode with a Clock Prescaler Factor of 1 261
Figure 14-7: Capture/Compare Channel (Input Section of Channel 1) 263
Figure 14-8: Main Circuit of Capture/Compare Channel 1 263
Figure 14-9: Output Section of Capture/Compare Channel 264
Figure 14-10: PWM Input Capture 266
Figure 14-11: Output Compare Mode, Toggle on OC1 267
Figure 14-12: Edge-aligned PWM Waveform (ARR = 7) 268
Figure 14-13: Center-aligned PWM Waveform (APR = 7) 269
Figure 14-14: Timing Diagram of Single-pulse Mode 271
Figure 14-15: GTIMER_ETR Signal Clearing OCxREF of GTIMER 273
Figure 14-16: Example of Counter Operation in Encoder Interface Mode 274
Figure 14-17: Timing Diagram in Reset Mode 275
Figure 14-18: Timing Diagram in Gated Mode 276
Figure 14-19: Timing Diagram in Trigger Mode 277
Figure 15-1: LPTIMER Block Diagram 302
Figure 21-1: Diagram of ADC Sampling via PGA Buffer 375
Figure 21-2: Flowchart of ADC Sampling via PGA Buffer 375
Figure 22-1: COMP Analog Functional Block Diagram 378
Figure 22-2: COMP Overall Functional Block Diagram 379

List of Tables

Table 3-1: Memory and Module Boundary Address 11
Table 5-1: System Clock Selection 15
Table 5-2: System Reset Sources 15
Table 5-3: Low-power Mode Summary 18
Table 5-4: List of System Registers 22
Table 6-1: List of EFC Registers 68
Table 7-1: Interrupt Source 80
Table 8-1: List of GPIO Registers 82
Table 9-1: List of UART0/1 Registers 88
Table 10-1: List of I2C Registers 94
Table 11-1: List of SPI Registers 111
Table 11-2: SPI Pin Configuration 120
Table 12-1: Input Signal Sources 144
Table 12-2: Counting Direction versus Encoder Signals 159
Table 12-3: DMA Access Counting Mode 165
Table 12-4: List of ATIMER Registers 167
Table 13-1: GTIMER0/1/2 Internal Signal Extension Table 214
Table 13-2: Encoder Counting Methods and Input Signals 225
Table 13-3: List of GTIMER Registers 231
Table 14-1: GTIMER0/1/2 Internal Signal Extension Table 262
Table 14-2: Encoder Counting Methods and Input Signals 273
Table 14-3: List of GTIMER Registers 279
Table 15-1: List of LPTIMER Registers 305
Table 17-1: List of CRC Registers 327
Table 18-1: List of RNG Registers 330
Table 19-1: List of WDT Registers 332
Table 20-1: List of WWDT Registers 337
Table 21-1: ADC Pin Description 342
Table 21-2: List of ADC Registers 343
Table 22-1: List of COMP Registers 380
Table 23-1: List of DIV Registers 398
Table 24-1: List of SQRT Registers 401
Table 25-1: List of SysTick Registers 403

1 Documentation Conventions

1.1 List of Abbreviations for Registers

The following abbreviations are used in register descriptions:

R/W	Read-write, software can read and write to this bit.
R	Read-only, software can only read this bit.
W	Write-only, software can only write to this bit, and reading this bit returns the reset value.
RC	Read-clear, reading clears this bit.
R/W1C	Read, software can read as well as clear this bit by writing 1. Writing 0 has no effect on the bit value.
R/W0C	Read, software can read this bit as well as clear this bit by writing 0. Writing 1 has no effect on the bit value.
RSV	Reserved bit, must be kept at reset value.

1.2 Glossary

This section outlines the definitions of acronyms and abbreviations used in this document:

- The CPU core integrates the SWD debug port (SWD-DP), providing a 2-pin (clock and data) interface based on the serial wire debug (SWD) protocol.
- Word: data/instruction of 32-bit length
- Half-word: data/instruction of 16-bit length
- Byte: data of 8-bit length
- Double-word: data of 64-bit length
- IAP (in-application programming): IAP is the ability to re-program the Flash memory of a microcontroller while the user program is running.

- ICP (in-circuit programming): ICP is the ability to program the Flash memory of a microcontroller using the JTAG protocol, the SWD protocol or the bootloader while the device is mounted on the user application board.
- Option bytes: production configuration bits stored in the Flash memory
- OBL: option byte loader
- AHB: advanced high-performance bus
- APB: advanced peripheral bus

2 Product Introduction

2.1 Overview

The UM32Mx56 series of chips is a 32-bit processor SoC series developed by Unicmicro based on the ARM Cortex-M0+ core. It features low pin count and a wide operating voltage range, and is primarily aimed at application scenarios such as automotive electrification, smart home, motor control, and industrial control. In accordance with the specific application requirements of the industry, the chip integrates general-purpose peripheral communication interfaces (including 12-bit SAR ADC, UART, SPI, I2C, etc.), sensor acquisition interfaces (including ADC, PGA, COMP, etc.), low-power module interfaces (including LPTIMER, WDT, etc.), and hardware algorithm modules (including SQRT, DIV, etc.). The chip features high integration, high interference immunity, high reliability and ultra-low power consumption. It has built-in RC oscillators of both high frequency and low frequency, supporting crystal-free applications. It also supports the Keil MDK integrated development environment and allows software development in C and assembly languages.

The UM32Mx56 series of chips has added security features to prevent data theft, including SWD read protection and physical disabling of the SWD interface. The internal eFlash memory of the chip has been enhanced with ECC error correction codes and data encryption capabilities. ECC error correction can effectively prevent data loss in eFlash or occasional read errors due to environmental factors. In addition, the chip has implemented a protection mechanism against erasure and rewriting for certain EEPROM areas and the 32KB eFlash main storage area, effectively preventing customer application data and programs from being rewritten.

Applications:

- Motor control

- Industrial IoT applications
- Intelligent transportation, smart cities, smart homes, etc.
- Intelligent sensor terminal applications such as smart door lock, asset tracking and wireless monitoring

2.2 Main Features

- **Ultra-low power management system**
  - Low power modules: LPTIMER, WDT
  - Built-in ROSC/LDO/POR, no crystal/LDO/ reset circuits required
- **Processor**
  - 32-bit ARM Cortex-M0+, up to 60MHz
  - Single-cycle hardware multiplier
- **Memory**
  - 32 KB FLASH
  - 4 KB EEPROM
  - 6 KB SRAM
  - Sector size: 512 B; endurance: 100,000 times
  - Data retention time: 10 years @ room temperature
- **Clock**
  - Internal RCH: 60 MHz
  - Internal RCL: 32 kHz
- **Analog peripherals**
  - 1 × 12-bit 1 Msps ADC, up to 16 channels
  - 3 x operational amplifiers
  - 2 x voltage comparators

- **Communication interfaces**
  - Up to 2 × UART serial ports
  - 1 × general-purpose SPI
  - 1 × I2C interface
- **Timers**
  - 1 × 16-bit ATIMER supporting input capture and complementary PWM output with dead-time insertion
  - 1 x 32-bit GTIMER and 2 x 16-bit GTIMERs supporting input capture and PWM output
  - 1 x 32-bit LPTIMER, supporting 2 PWM outputs
  - 1 x 32-bit low-powerWDT, resettable and interruptible
  - 1 x 18-bit WWDT, resettable and interruptible
  - Up to 21 PWM outputs in total
- **GPIO**
  - Up to 26 GPIOs
  - Supporting edge/level-triggered interrupts
  - Configurable driving capability: 16/8 mA
- **Security features**
  - Low-voltage detection (LVD)
  - Low-voltage reset (LVR), anti-crash
  - CRC16-CCITT hardware acceleration
  - Hardware random number generator (RNG)
  - 16-byte UUID
- **Hardware acceleration engine**
  - Divider (DIV), results in 8 clock cycles
  - Square root operation, results in 4 clock cycles

- **SIP pre-driver**
  - UM32xxx
    - ✓ Built-in 6 MOS drivers
    - ✓ Gate drive voltage: 8 V–20 V
    - ✓ Output-stage current: 1.5 A sourcing / 1.8 A sinking
  - UM32xxx
    - ✓ Built-in 6 PMOS + NMOS drivers
    - ✓ Built-in 5 V/40 mA LDO
    - ✓ Gate drive voltage: 5 V–30 V
    - ✓ Output-stage current: 50 mA sourcing / 300 mA sinking
- **Key electrical parameters**
  - Operating voltage: 2.3–5.5 V
  - Operating temperature: -40℃–105℃
  - ESD: ±8 kV (HBM)
- **Development support**
  - Updating of IAP applications
  - Support online debugging/download using JTAG-> SWD protocols
  - Complete SDK and EVB HDK
  - Off-line programmer

3 Memory and Bus Architecture

3.1 System Architecture

The main system consists of:

- Two AHB bus masters:
  - Cortex-M0+
  - DMA controller
- Five AHB bus slaves:
  - Flash memory
  - SRAM
  - AHB, AHB to APB bridge, including all APB and AHB interface peripherals
  - DIV divider
  - SQRT square root calculator

3.2 Bus Architecture Diagram

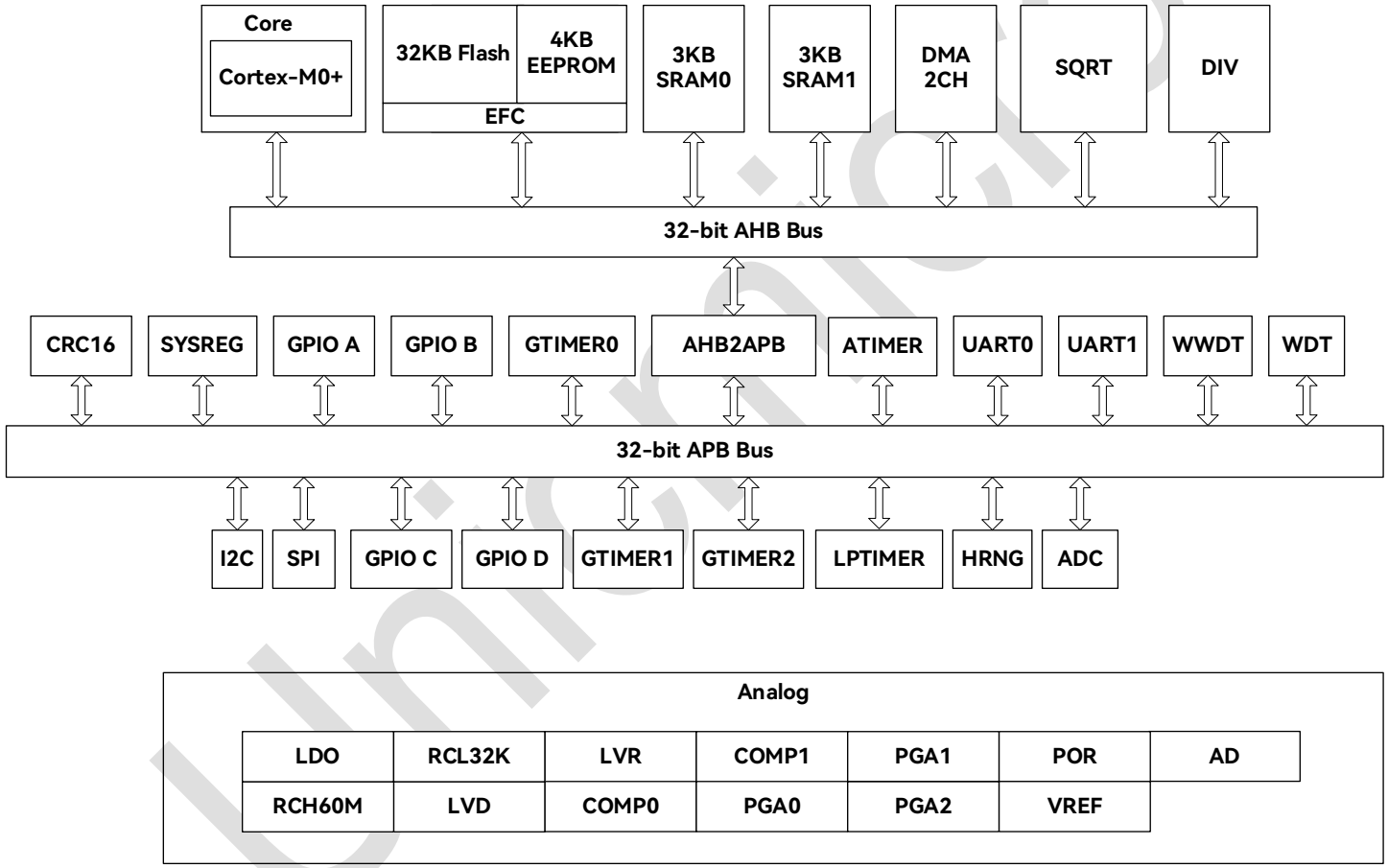


Figure 3-1: Bus Architecture Diagram

3.3 Memory Mapping

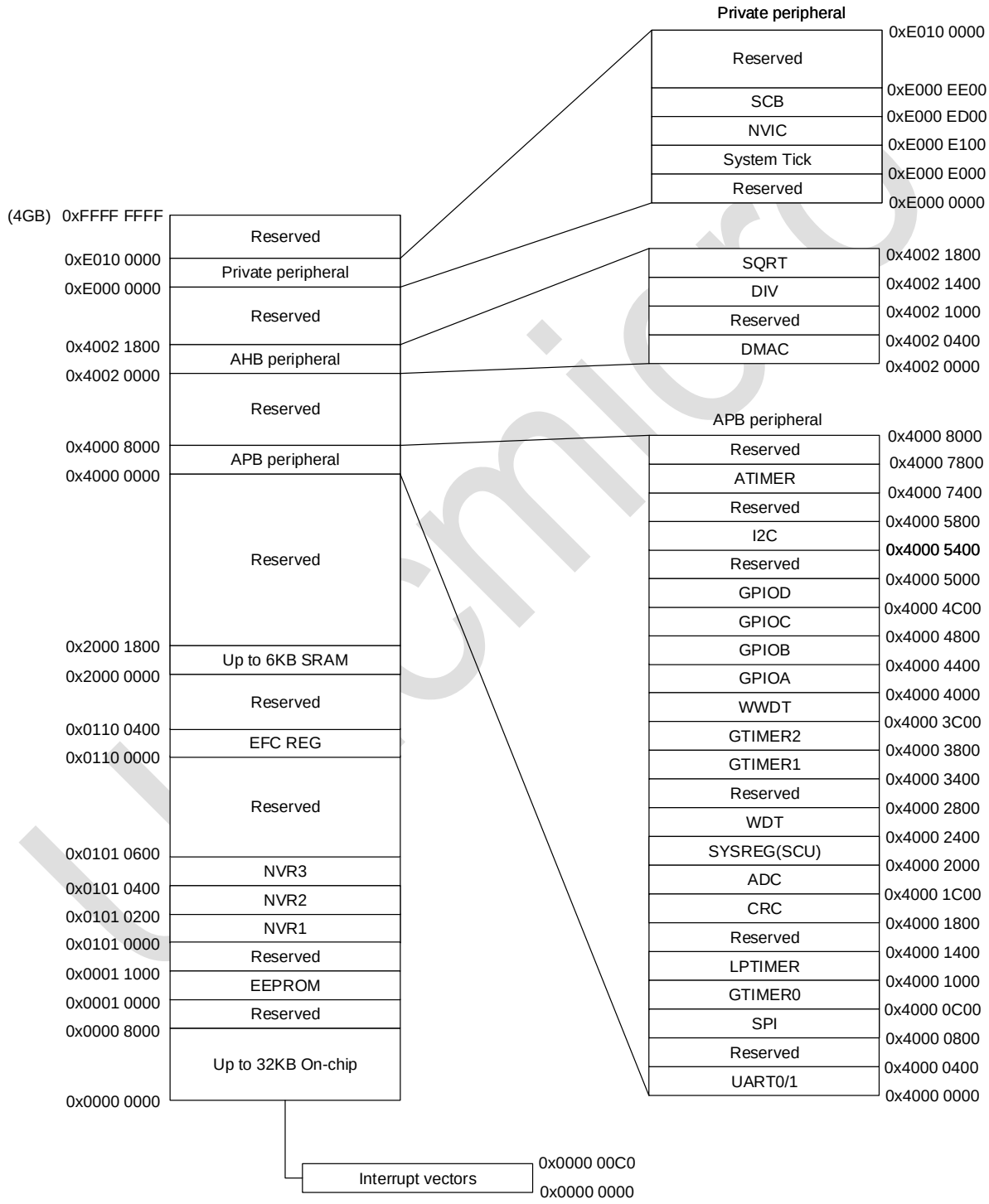


Figure 3-2: Memory Address Mapping Diagram

Table 3-1: Memory and Module Boundary Address

Module Name	Boundary Address	Size
EFlash	0x0000_0000 – 0x0000_8000	32 KB
EEPROM	0x0001_0000 – 0x0001_1000	4 KB
NVR1	0x0101_0000 – 0x0101_0200	512 B
NVR2	0x0101_0200 – 0x0101_0400	512 B
NVR3	0x0101_0400 – 0x0101_0600	512 B
EFC register	0x0110_0000 – 0x0110_006C	108 B
SRAM1	0x2000_0000 – 0x2000_0C00	3 KB
SRAM2	0x2000_0C00 – 0x2000_1800	3 KB
UART0/1	0X4000_0000 – 0x4000_0400	1 KB
SPI	0X4000_0800 – 0x4000_0C00	1 KB
GTIMER0	0x4000_0C00 – 0x4000_1000	1 KB
LPTIMER	0x4000_1000 – 0x4000_1400	1 KB
CRC	0x4000_1800 – 0x4000_1C00	1 KB
ADC	0x4000_1C00 – 0x4000_2000	1 KB
SYSREG (SCU)	0x4000_2000 – 0x4000_2400	1 KB
WDT	0x4000_2400 – 0x4000_2800	1 KB
GTIMER1	0x4000_3400 – 0x4000_3800	1 KB
GTIMER2	0x4000_3800 – 0x4000_3C00	1 KB
WWDT	0x4000_3C00 – 0x4000_4000	1 KB
GPIOA	0x4000_4000 – 0x4000_4400	1 KB
GPIOB	0x4000_4400 – 0x4000_4800	1 KB
GPIOC	0x4000_4800 – 0x4000_4C00	1 KB
GPIOD	0x4000_4C00 – 0x4000_5000	1 KB
I2C	0x4000_5400 – 0x4000_5800	1 KB
ATIMER	0x4000_7400 – 0x4000_7800	1 KB
DMAC	0x4002_0000 – 0x4002_0C00	3 KB
DIV	0x4002_1000 – 0x4002_1400	1 KB
SQRT	0x4002_1400 – 0x4002_1800	1 KB

4 Processor

4.1 Overview

The Cortex™ M0+ processor is a 32-bit two-stage pipelined RISC processor embedded with an AMBA-Lite interface and a nested vectored interrupt controller (NVIC). It features hardware debugging function, Thumb instruction execution and compatibility with other Cortex-M series processors. The processor also incorporates a number of brand-new designs as well as energy-saving and consumption-reducing technologies to improve debugging and tracing capabilities, reduce the number of instructions per cycle (IPC), and improve the two-stage pipeline for Flash access. The Cortex M0+ fully supports the integrated Keil & IAR debugger.

4.2 Main Features

- ARMv6-M Thumb
- Thumb/Thumb-2 technology
- ARMv6 M compatible with 24-bit SysTick timer
- 32-bit hardware multiplier
- Little-endian data access available in system interface
- Accurate and timely interrupt handling capability
- Instructions for loading and storing multiple data and multi-cycle multiplication can be terminated and then restarted, thus realizing quick interrupt handling.
- Exception-compatible mode for C application binary interface (C-ABI); ARMv6-M mode allows the user to implement interrupt handling using pure C function
- Executing wait-for-interrupt (WFI) and wait-for-event (WFE) instructions to enter low-power sleep mode or exit sleep mode from the interrupt.

4.3 Functional Block Diagram

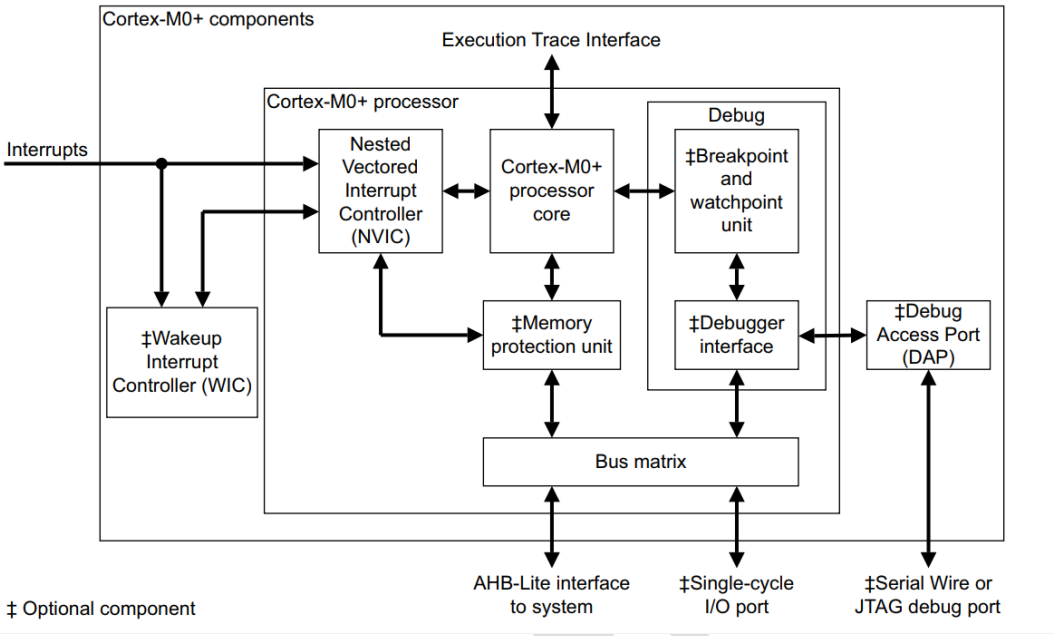


Figure 4-1: Cortex-M0+ Functional Block Diagram

4.4 Core Register Set

The register set of the Cortex-M0+ processor is shown in the figure below:

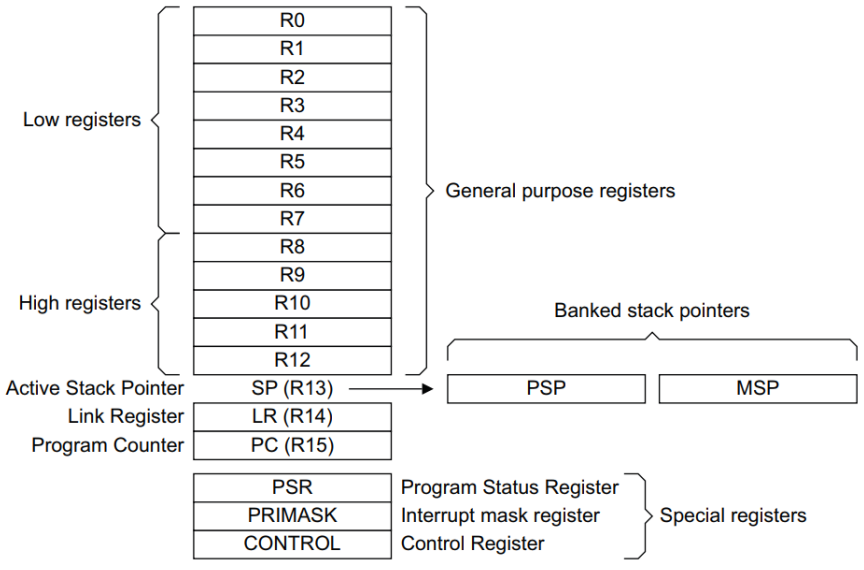


Figure 4-2: Cortex-M0+ Register Set

5 System Configuration Utility (SCU)

5.1 Clock Block Diagram

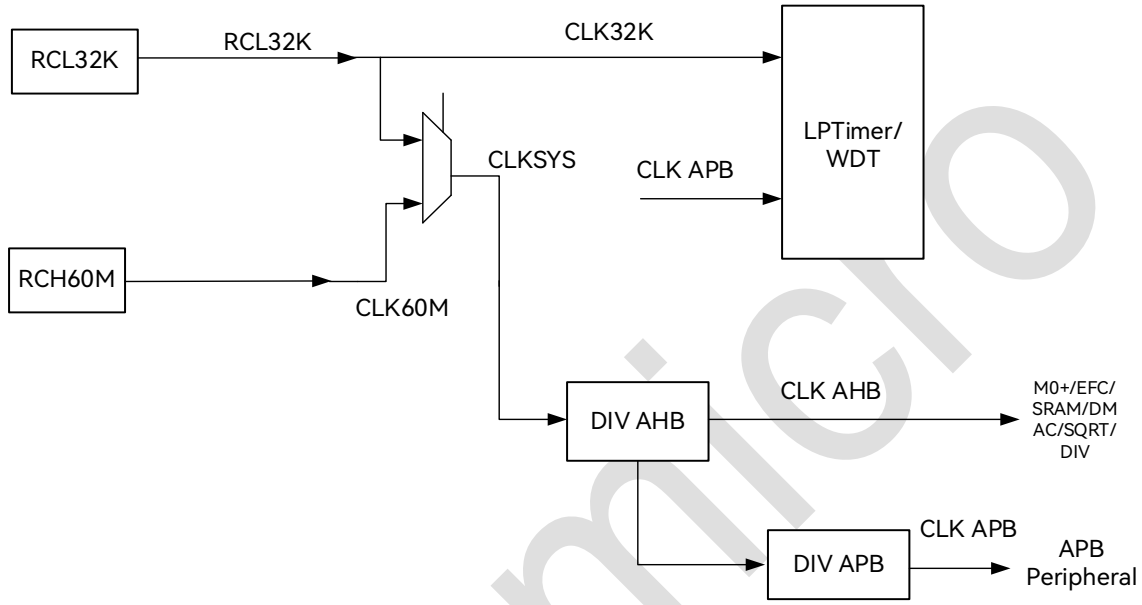


Figure 5-1: Clock Module Block Diagram

5.2 Clock Selection

The system has two clock sources:

- 60 MHz high-precision internal clock (RCH), serving as system clock source
- 32 kHz internal clock (RCL), serving as low-power clock and can also be used as system clock source

Different clock schemes are required for different operation modes. The system clock source can be selected via configuring the CLK_SEL bit [14] in the system control register 0 (SYSCTRL0). The relationship is shown in the table below:

Table 5-1: System Clock Selection

CLK_SEL	System Clock Source
0	RCH
CLK_SEL	**System Clock Source**
1	RCL

5.3 Reset Source

The chip has multiple reset sources, including POR, RESETEN, WDT reset, WWDT reset, SOFT_RESETN, module software reset, LVD reset and LOCKUP reset. The specific reset resources are listed below.

Table 5-2: System Reset Sources

Reset Source	Description
Internal analog POR	Reset everything
LVR	
RESETEN	Reset everything except the CPU DEBUG logic
LOCKUP Reset	Reset logic other than EFC
LVD reset	
WDT	
WWDT	
SOFT_RESETN	
Module reset	Reset the corresponding IP modules

5.3.1 Internal POR

Internal power-on reset (POR): unconditionally reset the entire chip.

5.3.2 LVR

Low-voltage reset: unconditionally reset the entire chip.

5.3.3 RESETEN

External reset RESETEN resets the entire chip except the CPU DEBUG logic. In RESETEN state, the chip can be connected to the SWD interface. The RESETEN pin defaults to an external reset after power-up, and can be disabled via software.

5.3.4 LOCKUP Reset

When the system experiences two consecutive hard faults, the CPU will enter the LOCKUP state and the system will generate a reset. The LOCKUP reset affects all logic except the EFC.

5.3.5 LVD Reset

The LVD reset affects all logic except the EFC.

5.3.6 WDT Reset

The watchdog timer, released by default, resets the entire user circuit except the EFC.

The WDT reset occurs when the software fails to effectively prevent a timeout event. This reset only happens when the software fails to execute properly and may corrupt data. When the CPU is in the HALT state, WDT stops counting and will not generate a reset signal.

5.3.7 WWDT Reset

The window watchdog timer (WWDT) reset affects all logic except the EFC.

5.3.8 SOFT_RESETN

This reset is generated by the system, which can be restarted by initiating a soft reset without resetting the EFC controller and IO-related settings.

5.3.9 Module Reset

Reset each digital module by software.

5.4 Low-power Mode

In addition to the normal operating mode, the chip provides three low-power modes to reduce current consumption: Sleep mode, Deepsleep mode and Stop mode.

In Sleep mode, CM0+ stops working while retaining the interrupt handling capabilities. The clocks and resets of other peripherals and modules can be set by software. Sleep mode is accessed by executing the CM0+ specific instructions WFI/WFE, and wake-up is triggered by interrupts.

Deepsleep mode is an upgraded version of Sleep mode, in which CM0+ and the high-speed clock stop running while the low-power functional modules (LPTIMER and WDT) can continue to operate. To enter DeepSleep mode, the DEEPSLEEP register inside the CM0+ must be set first, followed by the CM0+ specific instructions WFI/WFE. Wake-up is triggered by interrupts.

In Stop mode, both the high-speed and low-speed clocks are stopped, and there are no running clocks in the system. All peripheral modules are suspended. The power-on reset signal is effective. The IO status is maintained, and the IO interrupts are enabled. The power consumption when all registers, RAM and CPU data are preserved is as follows: To enter the stop mode, the STOPMODE_SEL register in the system register and the DEEPSLEEP register inside the CM0+ must be configured first. Then, the CM0+ specific instructions WFI/WFE are executed to enter the mode. Wake-up can be achieved through GPIO edge/level wake-up or through an external asynchronous pulse count interrupt generated by the LPTIMER.

Detailed descriptions are listed in the table below:

Table 5-3: Low-power Mode Summary

Mode	Mode Description	Entry Condition	Exit Condition
Sleep	Powered by LDO Active, the CPU is mostly in sleep mode (including NVIC), while WIC is not in sleep mode; software can disable the clocks of various modules.	1. Disable the clocks of peripheral modules as required, leaving only those needed for monitoring interrupt events. 2. Execute WFI/WFE instructions.	1. CM0+ detects an interrupt or event. 2. Enter the interrupt service routine to clear the interrupt and return. 3. Continue executing subsequent instructions.
DeepSleep	Powered by LDO Standby, the CPU is mostly in sleep mode (including NVIC), while WIC is not in sleep mode; the high-speed clock source is disabled, and the RCL low-speed clock source is running.	1. Disable the clocks of peripheral modules as required, leaving only those needed for monitoring interrupt events. 2. Set the DeepSleep register inside CM0+. 3. Execute WFI/WFE instructions.	1. CM0+ detects an interrupt or event. 2. Enter the interrupt service routine to clear the interrupt and return. 3. Continue executing subsequent instructions.
Stop	Powered by LDO Standby, all system clocks are disabled.	1. Set the conditions for IO wake-up as required. 2. Set the DeepSleep register inside CM0+. 3. Set the STOPMODE_SEL register in system register. 4. Execute WFI/WFE instructions.	1. An external IO wake-up event occurs. 2. CM0+ detects the interrupt triggered by external IO wake-up event. 3. Enter the interrupt service routine to clear the interrupt and return. 4. Continue executing subsequent instructions.

The entry and wake-up conditions of the low-power modes are summarized as follows:

- The entry conditions for Sleep, DeepSleep, and Stop modes all require the invocation of WFI/WFE; the essence of waking up from all three modes is the occurrence of an interrupt or event.
- In Sleep mode, the high-speed clock RCH and the internal low-speed clock RCL maintain their settings prior to entering the low-power mode, so that the system can wake up and exit upon the generation of an interrupt.
- In DeepSleep mode, the RCH clock is disabled, and the RCL clock operates according to its settings. Therefore, only modules working with the low-frequency clock source, such as WDT and LPTIMER, can generate interrupts to wake up and exit, and in GPIO edge/level mode, interrupts can be generated to wake up and exit without a clock.
- In Stop mode, all clock sources are disabled. Wake-up and exit can be achieved through GPIO edge/level mode generating interrupts without a clock, or through an external asynchronous pulse count interrupt generated by the LPTIMER.

5.4.1 Sleep Mode

Eentry conditions for Sleep mode:

- Configure bit 2 of SCB->SCR to 0.
- Execute WFI/WFE instructions to enter Sleep mode.

  Wake-up condition: interrupt

5.4.2 DeepSleep Mode

In DeepSleep mode, the wake-up sources include interrupts generated by WDT and LPTIMER, and GPIO edge/level interrupts.

The following is an example of the configuration process for waking up from a falling edge on

pin PD3:

1. Configure the peripheral module clock control register PERI_CLKEN [19] to enable the GPIOD clock.
2. Configure the peripheral module reset control register PERI_RESET[19] to set GPIOD to normal operation.
3. Configure the port PD function register PD_SEL[15:12] to set PD3 as GPIO function.
4. Configure the GPIOD data direction register GPIO_DIR[3] to set PD3 as an input.
5. Configure the GPIOD interrupt trigger mode register GPIO_IS[3] to set PD3 for edge-triggered interrupt.
6. Configure the GPIOD interrupt edge trigger setting register GPIO_IBE[3] to set PD3 for single-edge trigger.
7. Configure the GPIOD interrupt high/low level trigger setting register GPIO_IEV to set for falling-edge trigger.
8. Configure the GPIOD interrupt enable register GPIO_IEN[3] to enable the PD3 interrupt.
9. Configure the port input configuration register PAD_IE[27] to enable input on PD3.
10. Configure the port pull-up configuration register PAD_PU[27] to enable pull-up on PD3.
11. Configure SCB->SCR[2] to 1.
12. Invoke WFI to enter Deepsleep mode.
13. After an interrupt is generated, the system can be woken up from the DeepSleep mode.

5.4.3 Stop Mode

In Stop mode, the wake-up sources include GPIO edge/level interrupts or interrupts generated by the external asynchronous pulse count of LPTIMER.

The following is an example of the configuration process for waking up from a falling edge on

pin PD3:

1. Configure the peripheral module clock control register PERI_CLKEN [19] to enable the GPIOD clock.
2. Configure the peripheral module reset control register PERI_RESET[19] to set GPIOD to normal operation.
3. Configure the port PD function register PD_SEL[15:12] to set PD3 as GPIO function.
4. Configure the GPIOD data direction register GPIO_DIR[3] to set PD3 as an input.
5. Configure the GPIOD interrupt trigger mode register GPIO_IS[3] to set PD3 for edge-triggered interrupt.
6. Configure the GPIOD interrupt edge trigger setting register GPIO_IBE[3] to set PD3 as for single-edge trigger.
7. Configure the GPIOD interrupt high/low level trigger setting register GPIO_IEV to set for falling-edge trigger.
8. Configure the GPIOD interrupt enable register GPIO_IEN[3] to enable the PD3 interrupt.
9. Configure the port input configuration register PAD_IE[27] to enable input on PD3.
10. Configure the port pull-up configuration register PAD_PU[27] to enable pull-up on PD3.
11. Configure STOPMODE_SEL = 0xA5A50001; // Stop mode is enabled.
12. Configure SCB->SCR[2] to 1.
13. Invoke __WFI(); // Enter Stop mode.
14. After an interrupt is generated, the system can be woken up from Stop mode.

5.5 System Register

SYSREG (SCU) register base address: 0x4000_2000

Table 5-4: List of System Registers

Offset	Name	Description
0x000	SCU_CTRL0	System control register 0
0x008	SCU_CTRL_PROT	System control protection register
0x00C	SCU_OSC_CTRL	Clock control register
0x010	SCU_PERI_CLKEN	Peripheral clock gating register
0x020	SCU_RESET_FLAG	Reset flag register
0x024	SCU_PERI_RESET	Peripheral module reset control register
0x028	SCU_EXTRST_CTRL	External reset filter control register
0x030	SCU_PA_SEL	Port PA function configuration register, can only be reset externally or via POR
0x034	SCU_PB_SEL	Port PB function configuration register, can only be reset externally or via POR
0x038	SCU_PC_SEL	Port PC function configuration register, can only be reset externally or via POR
0x03C	SCU_PD_SEL	Port PD function configuration register, can only be reset externally or via POR
0x054	SCU_PAD_ADS	Port analog-digital configuration register, can only be reset externally or via POR
0x060	SCU_PAD_DR	Port drive capability configuration register, can only be reset externally or via POR
0x06C	SCU_PAD_PU	Port pull-up configuration register, can only be reset externally or via POR
0x078	SCU_PAD_PD	Port pull-down configuration register, can only be reset externally or via POR
0x084	SCU_PAD_OD	Port open-drain output configuration register, can only be reset externally or via POR
0x090	SCU_PAD_CS	Port input type configuration register, can only be reset externally or via POR
0x09C	SCU_PAD_IE	Port input enable register, can only be reset externally or via POR
0x0A4	SCU_PAD_STATUS	Port input level register
0x0A8	SCU_PAD_SR	Port speed configuration register, can only be reset

Offset	Name	Description
		externally or via POR
0x0B4	SCU_IOCTRL_PROT	IO control protection register
0x0B8	SCU_LVD_CFG	LVD control register
0x0BC	SCU_PGA_CFG	PGA control register
0x0D0	SCU_EXTRST_SEL	External reset port selection register
0x0D4	SCU_STOP_SEL	Stop mode selection register
0x0D8	SCU_SOFT_RSTN	Software reset register
0x0DC	SCU_VET_OFFSET	Interrupt vector address remapping register
0x0FC	SCU_LVD_INTR	LVD interrupt register
0x188	SCU_VREF	VREF control register

5.5.1 System Control Register 0 SCU_CTRL0 (Offset: 000H)

Bit	Name	Attribute	Reset Value	Description
31:29	RSV	-	-	Reserved
28:27	CLKOUT_SEL	RW	0x0	When IO is used as a clock output, the output clock selection is as follows: 00: HCLK_OUT (system clock) 01: RCL clock 10/11: PCLK_OUT
26:17	RSV	-	-	Reserved
16	SWD_WACK_EN	RW	0x1	Enable bit for waking up the system in DeepSleep or Stop mode after connecting ULINK or JLINK: 1: Use NMI interrupt to wake up the system under the above conditions. 0: Do not wake up the system under the above conditions
15	RSV	-	-	Reserved
14	CLK_SEL	RW	0x0	System clock source selection: 0: High-speed clock CLK_SEL_HF 1: Low-speed clock CLK_SEL_LF
13:11	RSV	-	-	Reserved
10:9	PCLK_DIV	RW	0x0	PCLK divider selection: 00: HCLK 01: HCLK/2

Bit	Name	Attribute	Reset Value	Description
				10: HCLK/4 11: HCLK/8
8:6	HCLK_DIV	RW	0x3	HCLK divider selection: 000: SystemClk 001: SystemClk/2 010: SystemClk/4 011: SystemClk/8 100: SystemClk/16 101: SystemClk/32 110: SystemClk/64 111: SystemClk/128
5:3	RSV	-	-	Reserved
2	RCL_EN	RW	0x1	Internal low-speed clock (RCL) enable: 0: Disabled 1: Enabled
1	RSV	-	-	Reserved
0	RCH_EN	RW	0x1	Internal high-speed clock (RCH) enable: 0: Disabled 1: Enabled Note: When the system enters DeepSleep mode, this high-speed clock will be automatically disabled.

5.5.2 System Control Protection Register SCU_CTRL_PROT (Offset: 008H)

Bit	Name	Attribute	Reset Value	Description
31:0	SYSCTRL_PROTECT	RW	0x0	SCU_CTRL0 write protection control register: Writing 0xA5A5_5A5A to this register enables writing to the SCU_CTRL0 register. Writing any other value disables the write access. After configuring the SCU_CTRL0 register, write access will be automatically disabled. Reading this register will return the write

Bit	Name	Attribute	Reset Value	Description
				enable status of SCU_CTRL0 register. 0: Write not enabled 1: Write enabled

5.5.3 Clock Control Register SCU_OSC_CTRL (Offset: 0x00CH)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27:16	WAKP_delay	RW	0x25A	The time delay provided by the CLK after the system wakes up from DeepSleep/Stop mode. The time in this register is counted in system clock cycles.
15:0	RSV	-	-	Reserved

5.5.4 Peripheral Module Clock Register SCU_PERI_CLKEN (Offset: 010H)

Bit	Name	Attribute	Reset Value	Description
31	DIV_CLKEN	RW	0x0	DIV module clock enable: 1: Enabled 0: Disabled
30	UART1_CLKEN	RW	0x0	UART1 module clock enable: 1: Enabled 0: Disabled
29:28	RSV	-	-	Reserved
27	ATIMER_CLKEN	RW	0x0	ATIMER module clock enable: 1: Enabled 0: Disabled
26	SQRT_CLKEN	RW	0x0	SQRT module clock enable: 1: Enabled 0: Disabled
25	WWDT_CLKEN	RW	0x0	WWDT module clock enable: 1: Enabled 0: Disabled
24:23	RSV	-	-	Reserved
22	DMA_CLKEN	RW	0x0	DMA module clock enable: 1: Enabled 0: Disabled
21:20	RSV	-	-	Reserved
19	GPIOD_CLKEN	RW	0x0	GPIOD module clock enable: 1: Enabled 0: Disabled

Bit	Name	Attribute	Reset Value	Description
18	GPIOC_CLKEN	RW	0x0	GPIOC module clock enable: 1: Enabled 0: Disabled
17	GPIOB_CLKEN	RW	0x0	GPIOB module clock enable: 1: Enabled 0: Disabled
16	GPIOA_CLKEN	RW	0x0	GPIOA module clock enable: 1: Enabled 0: Disabled
15	I2C_CLKEN	RW	0x0	I2C module clock enable: 1: Enabled 0: Disabled
14	ADC_CLKEN	RW	0x0	ADC module clock enable: 1: Enabled 0: Disabled
13	RSV	-	-	Reserved
12	WDT_CLKEN	RW	0x0	WDT module clock enable: 1: Enabled 0: Disabled
11	CRC_CLKEN	RW	0x0	CRC16 module clock enable: 1: Enabled 0: Disabled
10	RSV	-	-	Reserved
9	GTIMER0_CLKEN	RW	0x0	GTIMER0 module clock enable: 1: Enabled 0: Disabled
8	LPTIMER_CLKEN	RW	0x0	LPTIMER module clock enable: 1: Enabled 0: Disabled
7:5	RSV	-	-	Reserved
4	SPI_CLKEN	RW	0x0	SPI module clock enable: 1: Enabled 0: Disabled
3	GTIMER2_CLKEN	RW	0x0	GTIMER 2 module clock enable: 1: Enabled 0: Disabled
2	GTIMER1_CLKEN	RW	0x0	GTIMER 1 module clock enable: 1: Enabled 0: Disabled
1	RSV	-	-	Reserved
0	UART0_CLKEN	RW	0x0	UART0 module clock enable: 1: Enabled 0: Disabled

5.5.5 Reset Flag Register SCU_RESET_FLAG (Offset: 020H)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
7	SYS_RSTREQ_FLAG	W1C	0x0	CPU reset status, requiring software initialization and clearing: 1: Cortex M0+ system reset occurs 0: No reset occurs This bit can only be reset by PORN and external RESETN, and can be cleared by writing 1.
6	LOCKUP_RSTN_FLAG	W1C	0x0	CPU lockup reset status, requiring software initialization and clearing: 1: Lockup reset occurs 0: No reset occurs This bit can only be reset by PORN and external RESETN, and can be cleared by writing 1.
5	LVD_RSTN_FLAG	W1C	0x0	Low-voltage reset status, requiring software initialization and clearing: 1: LVD reset occurs 0: No reset occurs This bit can only be reset by PORN and external RESETN, and can be cleared by writing 1.
4	WWDT_FLAG	W1C	0x0	Window watchdog reset status, requiring software initialization and clearing: 1: WWDT reset occurs 0: No reset occurs This bit can only be reset by PORN and external RESETN, and can be cleared by writing 1.
3	WDT_FLAG	W1C	0x0	Watchdog reset status, requiring software initialization and clearing: 1: WDT reset occurs 0: No reset occurs This bit can only be reset by PORN and external RESETN, and can be cleared by writing 1.

Bit	Name	Attribute	Reset Value	Description
2	RESETN_FLAG	W1C	0x0	External reset status, requiring software initialization and clearing: 1: External reset occurs. 0: No reset occurs This bit can only be reset by PORN, and can be cleared by writing 1.
1	SOFTR_FLAG	W1C	0x0	Soft reset status, requiring software initialization and clearing: 1: Soft reset occurs 0: No reset occurs This bit can only be reset by PORN and external RESETN, and can be cleared by writing 1.
0	RSV	-	-	Reserved

Note: The reset flag register can only be reset by POR.

5.5.6 Peripheral Module Reset Control Register SCU_PERI_RESET (Offset: 024H)

Bit	Name	Attribute	Reset Value	Description
31	DIV_RESET	RW	0x0	DIV module reset enable: 1: Module in normal operation 0: Module in reset state
30	UART1_RESET	RW	0x0	UART1 module reset enable: 1: Module in normal operation 0: Module in reset state
29:28	RSV	-	-	Reserved
27	ATIMER_RESET	RW	0x0	ATIMER module reset enable: 1: Module in normal operation 0: Module in reset state
26	SQRT_RESET	RW	0x0	SQRT module reset enable: 1: Module in normal operation 0: Module in reset state

Bit	Name	Attribute	Reset Value	Description
25	WWDT_RESET	RW	0x0	WWDT module reset enable: 1: Module in normal operation 0: Module in reset state
24:23	RSV	-	-	Reserved
22	DMA_RESET	RW	0x0	DMA module reset enable: 1: Module in normal operation 0: Module in reset state
21:20	RSV	-	-	Reserved
19	GPIOD_RESET	RW	0x0	GPIOD module reset enable: 1: Module in normal operation 0: Module in reset state
18	GPIOC_RESET	RW	0x0	GPIOC module reset enable: 1: Module in normal operation 0: Module in reset state
17	GPIOB_RESET	RW	0x0	GPIOB module reset enable: 1: Module in normal operation 0: Module in reset state
16	GPIOA_RESET	RW	0x0	GPIOA module reset enable: 1: Module in normal operation 0: Module in reset state
15	I2C_RESET	RW	0x0	I2C module reset enable: 1: Module in normal operation 0: Module in reset state
14	ADC_RESET	RW	0x0	ADC module reset enable: 1: Module in normal operation 0: Module in reset state
13	RSV	-	-	Reserved
12	WDT_RESET	RW	0x0	WDT module reset enable: 1: Module in normal operation 0: Module in reset state
11	CRC_RESET	RW	0x0	CRC16 module reset enable: 1: Module in normal operation 0: Module in reset state
10	RSV	-	-	Reserved
9	GTIMER0_RESET	RW	0x0	GTIMER0 module reset enable: 1: Module in normal operation 0: Module in reset state

Bit	Name	Attribute	Reset Value	Description
8	LPTIMER_RESET	RW	0x0	LPTIMER module reset enable: 1: Module in normal operation 0: Module in reset state
7:5	RSV	-	-	Reserved
4	SPI_RESET	RW	0x0	SPI module reset enable: 1: Module in normal operation 0: Module in reset state
3	GTIMER2_RESET	RW	0x0	GTIMER2 module reset enable: 1: Module in normal operation 0: Module in reset state
2	GTIMER1_RESET	RW	0x0	GTIMER1 module reset enable: 1: Module in normal operation 0: Module in reset state
1	RSV	-	-	Reserved
0	UART0_RESET	RW	0x0	UART0 module reset enable: 1: Module in normal operation 0: Module in reset state

5.5.7 External Reset Filter Control Register SCU_EXTRST_CTRL (Offset: 028H)

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	EXT_FILTER_EN	RW	0x0	External reset filter enable: 1: Enabled 0: Disabled

5.5.8 Port PA Function Configuration Register SCU_PA_SEL (Offset: 030H)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27:24	PA6_SEL	RW	0x0	Port PA6 function selection: 4'b0000: GPIO PA6

Bit	Name	Attribute	Reset Value	Description
				4’b0001: GTIM1_CH1 4’b0010: GTIM1_ETR 4’b0011: GTIM2_CH1 4’b0100: SPI_SCK 4’b0101: ATIM_CH1 4’b0110: ATIM_CH1N 4’b0111: ATIM_CH2 4’b1000: ATIM_CH2N 4’b1001: ATIM_CH3 4’b1010: ATIM_CH3N 4’b1011: ATIM_CH4
23:20	PA5_SEL	RW	0x0	Port PA5 function selection: 4’b0000: GPIO PA5 4’b0001: GTIM1_CH2 4’b0010: GTIM2_CH2 4’b0011: SPI_CSN0 4’b0100: ATIM_CH1 4’b0101: ATIM_CH1N 4’b0110: ATIM_CH2 4’b0111: ATIM_CH2N 4’b1000: ATIM_CH3 4’b1001: ATIM_CH3N 4’b1010: ATIM_CH4 4’b1011: ATIM_BK0
19:16	PA4_SEL	RW	0x0	Port PA4 function selection: 4’b0000: GPIO PA4 4’b0001: UART1_RX 4’b0010: GTIM1_CH2 4’b0011: SPI_MISO 4’b0100: ATIM_CH1 4’b0101: ATIM_CH1N 4’b0110: ATIM_CH2 4’b0111: ATIM_CH2N 4’b1000: ATIM_CH3 4’b1001: ATIM_CH3N 4’b1010: ATIM_CH4 4’b1011: LPTIM_CAP2

Bit	Name	Attribute	Reset Value	Description
15:12	PA3_SEL	RW	0x0	Port PA3 function selection: 4’b0000: GPIO PA3 4’b0001: UART1_TX 4’b0010: GTIM0_CH3 4’b0011: GTIM1_CH4 4’b0100: ATIM_CH1 4’b0101: ATIM_CH1N 4’b0110: ATIM_CH2 4’b0111: ATIM_CH2N 4’b1000: ATIM_CH3 4’b1001: ATIM_CH3N 4’b1010: ATIM_CH4 4’b1011: LVD_OUT
11:8	PA2_SEL	RW	0x0	Port PA2 function selection: 4’b0000: GPIO PA2 4’b0001: NC 4’b0010: GTIM0_CH1 4’b0011: GTIM1_CH2 4’b0100: GTIM2_CH1 4’b0101: GTIM2_CH3 4’b0110: SPI_MOSI 4’b0111: ATIM_ETR 4’b1000: LPTIM _ETR 4’b1001: LPTIM _CAP1 4’b1010: I2C_SCL 4’b1011: I2C_SDA
7:4	PA1_SEL	RW	0x0	Port PA1 function selection: 4’b0000: GPIO PA1 4’b0001: UART0_RX 4’b0010: GTIM0_CH2 4’b0011: GTIM1_CH3 4’b0100: GTIM2_ETR 4’b0101: ATIM_CH1 4’b0110: ATIM_CH1N 4’b0111: ATIM_CH2 4’b1000: ATIM_CH2N 4’b1001: ATIM_CH3

Bit	Name	Attribute	Reset Value	Description
				4’b1010: ATIM_CH3N 4’b1011: ATIM_CH4
3:0	PA0_SEL	RW	0x0	Port PA0 function selection: 4’b0000: GPIO PA0 4’b0001: UART0_TX 4’b0010: GTIM0_CH1 4’b0011: GTIM1_CH2 4’b0100: GTIM1_ETR 4’b0101: ATIM_CH1 4’b0110: ATIM_CH1N 4’b0111: ATIM_CH2 4’b1000: ATIM_CH2N 4’b1001: ATIM_CH3 4’b1010: ATIM_CH3N 4’b1011: ATIM_CH4

5.5.9 Port PB Function Configuration Register SCU_PB_SEL (Offset: 034H)

Bit	Name	Attribute	Reset Value	Description
31:28	PB7_SEL	RW	0x0	Port PB7 function selection: 4’b0000: GPIO PB7 4’b0001: UART1_RX 4’b0010: GTIM0_CH2 4’b0011: GTIM0_CH3 4’b0100: GTIM0_ETR 4’b0101: GTIM1_CH2 4’b0110: GTIM1_CH3 4’b0111: GTIM1_ETR 4’b1000: GTIM2_CH3 4’b1001: GTIM2_ETR 4’b1010: SPI_CSN0 4’b1011: SPI_MOSI 4’b1100: ATIM_ETR 4’b1101: ATIM_BK4 4’b1110: LPTIM _ETR

Bit	Name	Attribute	Reset Value	Description
				4’b1111: I2C_SCL
27:24	PB6_SEL	RW	0x0	Port PB6 function selection: 4’b0000: GPIO PB6 4’b0001: GTIM0_CH1 4’b0010: GTIM0_CH4 4’b0011: GTIM1_CH3 4’b0100: GTIM2_CH1 4’b0101: GTIM2_CH2 4’b0110: SPI_SCK 4’b0111: ATIM_ETR 4’b1000: LPTIM _OUT1 4’b1001: LPTIM _IN 4’b1010: LPTIM _CAP2 4’b1011: COMP0_OUT
23:20	PB5_SEL	RW	0x0	Port PB5 function selection: 4’b0000: GPIO PB5 4’b0001: UART1_RX 4’b0010: GTIM0_CH2 4’b0011: GTIM1_CH1 4’b0100: GTIM1_CH4 4’b0101: GTIM2_CH2 4’b0110: GTIM2_CH3 4’b0111: SPI_CSN1 4’b1000: SPI_MISO 4’b1001: SPI_MOSI 4’b1010: CLK_OUT 4’b1011: LPTIM _OUT2 4’b1100: COMP1_OUT
19:16	PB4_SEL	RW	0x0	Port PB4 function selection: 4’b0000: GPIO PB4 4’b0001: UART1_TX 4’b0010: GTIM0_CH3 4’b0011: GTIM0_ETR 4’b0100: GTIM1_CH2 4’b0101: GTIM1_ETR 4’b0110: GTIM2_CH3 4’b0111: GTIM2_ETR

Bit	Name	Attribute	Reset Value	Description
				4’b1000: SPI_CSN0 4’b1001: SPI_MISO 4’b1010: SPI_MOSI 4’b1011: ATIM_CH4 4’b1100: I2C_SDA
15:12	PB3_SEL	RW	0x0	Port PB3 function selection: 4’b0000: GPIO PB3 4’b0001: UART0_TX 4’b0010: UART0_RX 4’b0011: GTIM0_CH4 4’b0100: GTIM1_CH1 4’b0101: GTIM1_CH3 4’b0110: GTIM2_CH1 4’b0111: SPI_CSN1 4’b1000: SPI_SCK 4’b1001: SPI_MOSI 4’b1010: ATIM_CH1 4’b1011: ATIM_CH1N
11:8	PB2_SEL	RW	0x0	Port PB2 function selection: 4’b0000: GPIO PB2 4’b0001: GTIM0_CH1 4’b0010: GTIM0_CH3 4’b0011: GTIM0_ETR 4’b0100: GTIM2_CH1 4’b0101: GTIM2_CH4 4’b0110: SPI_MISO 4’b0111: SPI_MI1 4’b1000: ATIM_CH1 4’b1001: ATIM_CH1N 4’b1010: ATIM_CH2 4’b1011: ATIM_CH2N 4’b1100: ATIM_BK1 4’b1101: I2C_SCL
7:4	PB1_SEL	RW	0x0	Port PB1 function selection: 4’b0000: GPIO PB1 4’b0001: UART0_TX 4’b0010: GTIM0_CH2

Bit	Name	Attribute	Reset Value	Description
				4’b0011: GTIM0_ETR 4’b0100: GTIM1_CH3 4’b0101: GTIM2_CH2 4’b0110: SPI_CSN1 4’b0111: SPI_SCK 4’b1000: SPI_MOSI 4’b1001: ATIM_CH2N 4’b1010: ATIM_CH3 4’b1011: ATIM_CH3N 4’b1100: CLK_OUT 4’b1101: I2C_SDA
3:0	PB0_SEL	RW	0x0	Port PB0 function selection: 4’b0000: GPIO PB0 4’b0001: UART0_RX 4’b0010: GTIM0_CH1 4’b0011: GTIM0_ETR 4’b0100: GTIM1_CH2 4’b0101: GTIM1_ETR 4’b0110: GTIM2_CH3 4’b0111: SPI_CSN0 4’b1000: SPI_MOSI 4’b1001: ATIM_CH1 4’b1010: ATIM_CH1N 4’b1011: ATIM_CH2 4’b1100: ATIM_CH2N 4’b1101: ATIM_CH3N 4’b1110: I2C_SCL

5.5.10 Port PC Function Configuration Register SCU_PC_SEL (Offset: 038H)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27:24	PC6_SEL	RW	0x1	Port PC6 function selection: 4’b0000: GPIO PC6 4’b0001: SWCLK

Bit	Name	Attribute	Reset Value	Description
				4’b0010: UART0_TX 4’b0011: GTIM1_CH3 4’b0100: GTIM2_CH1 4’b0101: GTIM2_CH4 4’b0110: SPI_CSN0 4’b0111: SPI_MISO 4’b1000: SPI_MOSI 4’b1001: ATIM_ETR 4’b1010: I2C_SDA 4’b1011: COMP1_OUT
23:20	PC5_SEL	RW	0x1	Port PC5 function selection: 4’b0000: GPIO PC5 4’b0001: SWIO 4’b0010: UART0_RX 4’b0011: UART1_TX 4’b0100: GTIM1_CH1 4’b0101: GTIM2_CH2 4’b0110: GTIM2_CH3 4’b0111: SPI_SCK 4’b1000: SPI_MI1 4’b1001: CLK_OUT 4’b1010: I2C_SCL 4’b1011: COMP0_OUT
19:16	PC4_SEL	RW	0x0	Port PC4 function selection: 4’b0000: GPIO PC4 4’b0001: UART0_RX 4’b0010: GTIM0_CH1 4’b0011: GTIM1_CH2 4’b0100: GTIM1_CH3 4’b0101: GTIM2_CH3 4’b0110: GTIM2_ETR 4’b0111: SPI_MOSI 4’b1000: ATIM_CH2N 4’b1001: ATIM_CH3 4’b1010: ATIM_ETR 4’b1011: ATIM_BK5 4’b1100: ADC_TRO

Bit	Name	Attribute	Reset Value	Description
				4’b1101: I2C_SDA 4’b1110: COMP1_OUT
15:12	PC3_SEL	RW	0x0	Port PC3 function selection: 4’b0000: GPIO PC3 4’b0001: UART0_TX 4’b0010: GTIM0_CH2 4’b0011: GTIM0_ETR 4’b0100: GTIM1_CH1 4’b0101: GTIM1_CH3 4’b0110: GTIM1_CH4 4’b0111: GTIM2_CH1 4’b1000: GTIM2_CH3 4’b1001: ATIM_CH1N 4’b1010: ATIM_ETR 4’b1011: ADC_TRO 4’b1100: LPTIM _IN 4’b1101: LPTIM _CAP2 4’b1110: CAMP0_OUT
11:8	PC2_SEL	RW	0x0	Port PC2 function selection: 4’b0000: GPIO PC2 4’b0001: UART0_RX 4’b0010: GTIM0_CH1 4’b0011: GTIM0_CH3 4’b0100: GTIM0_ETR 4’b0101: GTIM1_CH2 4’b0110: GTIM1_CH3 4’b0111: GTIM1_ETR 4’b1000: GTIM2_CH4 4’b1001: ATIM_ETR 4’b1010: ATIM_BK6 4’b1011: ADC_TRO 4’b1100: COMP1_OUT
7:4	PC1_SEL	RW	0x0	Port PC1 function selection: 4’b0000: GPIO PC1 4’b0001: UART1_TX 4’b0010: GTIM0_CH1 4’b0011: GTIM0_CH4

Bit	Name	Attribute	Reset Value	Description
				4’b0100: GTIM1_CH4 4’b0101: GTIM2_CH2 4’b0110: GTIM2_ETR 4’b0111: SPI_SCK 4’b1000: LVD_OUT 4’b1001: LPTIM _OUT2 4’b1010: LPTIM _IN 4’b1011: LPTIM _CAP2
3:0	PC0_SEL	RW	0x0	Port PC0 function selection: 4’b0000: GPIO PC0 4’b0001: UART1_RX 4’b0010: GTIM0_CH2 4’b0011: GTIM1_CH1 4’b0100: GTIM2_CH2 4’b0101: GTIM2_CH3 4’b0110: SPI_CSN1 4’b0111: SPI_MISO 4’b1000: SPI_MI1 4’b1001: LPTIM _OUT1 4’b1010: LPTIM _CAP1 4’b1011: I2C_SDA

5.5.11 Port PD Function Configuration Register SCU_PD_SEL (Offset: 03CH)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	PD3_SEL	RW	0x0	Port PD3 function selection: 4’b0000: GPIO PD3 4’b0001: UART1_RX 4’b0010: GTIM0_CH2 4’b0011: GTIM2_CH1 4’b0100: SPI_MOSI 4’b0101: ATIM_CH1 4’b0110: ATIM_CH1N 4’b0111: ATIM_CH2

Bit	Name	Attribute	Reset Value	Description
				4’b1000: ATIM_CH2N 4’b1001: ATIM_CH3 4’b1010: ATIM_CH3N 4’b1011: ATIM_CH4 4’b1100: ATIM_ETR 4’b1101: ATIM_BK2 4’b1110: ADC_TRO 4’b1111: I2C_SDA
11:8	PD2_SEL	RW	0x0	Port PD2 function selection: 4’b0000: GPIO PD2 4’b0001: UART1_TX 4’b0010: GTIM0_CH3 4’b0011: GTIM0_ETR 4’b0100: GTIM2_CH1 4’b0101: GTIM2_CH2 4’b0110: GTIM2_CH3 4’b0111: SPI_SCK 4’b1000: SPI_MOSI 4’b1001: ATIM_CH1 4’b1010: ATIM_CH1N 4’b1011: ATIM_CH2 4’b1100: ATIM_CH2N 4’b1101: ATIM_CH3 4’b1110: ATIM_CH3N 4’b1111: I2C_SCL
7:4	PD1_SEL	RW	0x0	Port PD1 function selection: 4’b0000: GPIO PD1 4’b0001: UART0_TX 4’b0010: UART1_RX 4’b0011: GTIM0_CH1 4’b0100: GTIM0_ETR 4’b0101: GTIM1_ETR 4’b0110: GTIM2_CH2 4’b0111: SPI_MISO 4’b1000: SPI_MI1 4’b1001: ATIM_CH3 4’b1010: ATIM_CH3N

Bit	Name	Attribute	Reset Value	Description
				4’b1011: ATIM_ETR 4’b1100: ADC_TRO 4’b1101: I2C_SDA 4’b1110: COMP0_OUT
3:0	PD0_SEL	RW	0x0	Port PD0 function selection: 4’b0000: GPIO PD0 4’b0001: UART0_TX 4’b0010: GTIM0_CH2 4’b0011: GTIM0_CH3 4’b0100: GTIM0_ETR 4’b0101: GTIM1_CH1 4’b0110: SPI_CSN0 4’b0111: SPI_CSN1 4’b1000: SPI_SCK 4’b1001: ATIM_CH2N 4’b1010: ATIM_CH3 4’b1011: ATIM_ETR 4’b1100: ATIM_BK3 4’b1101: ADC_TRO 4’b1110: I2C_SCL 4’b1111: COMP1_OUT

5.5.12 Port Analog-digital Configuration Register SCU_PAD_ADS (Offset: 054H)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	PD3_ADS	RW	0x0	Port PD3 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
26	PD2_ADS	RW	0x0	Port PD2 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface

Bit	Name	Attribute	Reset Value	Description
25	PD1_ADS	RW	0x0	Port PD1 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
24	PD0_ADS	RW	0x0	Port PD0 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
23:21	RSV	-	-	Reserved
20	PC4_ADS	RW	0x0	Port PC4 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
19	PC3_ADS	RW	0x0	Port PC3 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
18	PC2_ADS	RW	0x0	Port PC2 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
17	PC1_ADS	RW	0x0	Port PC1 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
16	PC0_ADS	RW	0x0	Port PC0 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
15	PB7_ADS	RW	0x0	Port PB7 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
14	PB6_ADS	RW	0x0	Port PB6 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
13	PB5_ADS	RW	0x0	Port PB5 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
12	PB4_ADS	RW	0x0	Port PB4 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface

Bit	Name	Attribute	Reset Value	Description
11	PB3_ADS	RW	0x0	Port PB3 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
10	PB2_ADS	RW	0x0	Port PB2 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
9	PB1_ADS	RW	0x0	Port PB1 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
8	PB0_ADS	RW	0x0	Port PB0 analog-digital configuration register: 0: Configured as a digital interface 1: Configured as an analog interface
7:0	RSV	-	-	Reserved

5.5.13 Port Drive Capability Configuration Register SCU_PAD_DR (Offset: 060H)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	PD3_DR	RW	0x0	Port PD3 drive capability configuration register: 0: High drive capability; 1: Low drive capability
26	PD2_DR	RW	0x0	Port PD2 drive capability configuration register: 0: High drive capability; 1: Low drive capability
25	PD1_DR	RW	0x0	Port PD1 drive capability configuration register: 0: High drive capability; 1: Low drive capability
24	PD0_DR	RW	0x0	Port PD0 drive capability configuration register: 0: High drive capability; 1: Low drive capability
23	RSV	-	-	Reserved
22	PC6_DR	RW	0x0	Port PC6 drive capability configuration register: 0: High drive capability; 1: Low drive capability
21	PC5_DR	RW	0x0	Port PC5 drive capability configuration register: 0: High drive capability; 1: Low drive capability

Bit	Name	Attribute	Reset Value	Description
20	PC4_DR	RW	0x0	Port PC4 drive capability configuration register: 0: High drive capability; 1: Low drive capability
19	PC3_DR	RW	0x0	Port PC3 drive capability configuration register: 0: High drive capability; 1: Low drive capability
18	PC2_DR	RW	0x0	Port PC2 drive capability configuration register: 0: High drive capability; 1: Low drive capability
17	PC1_DR	RW	0x0	Port PC1 drive capability configuration register: 0: High drive capability; 1: Low drive capability
16	PC0_DR	RW	0x0	Port PC0 drive capability configuration register: 0: High drive capability; 1: Low drive capability
15	PB7_DR	RW	0x0	Port PB7 drive capability configuration register: 0: High drive capability; 1: Low drive capability
14	PB6_DR	RW	0x0	Port PB6 drive capability configuration register: 0: High drive capability; 1: Low drive capability
13	PB5_DR	RW	0x0	Port PB5 drive capability configuration register: 0: High drive capability; 1: Low drive capability
12	PB4_DR	RW	0x0	Port PB4 drive capability configuration register: 0: High drive capability; 1: Low drive capability
11	PB3_DR	RW	0x0	Port PB3 drive capability configuration register: 0: High drive capability; 1: Low drive capability
10	PB2_DR	RW	0x0	Port PB2 drive capability configuration register: 0: High drive capability; 1: Low drive capability
9	PB1_DR	RW	0x0	Port PB1 drive capability configuration register: 0: High drive capability; 1: Low drive capability
8	PB0_DR	RW	0x0	Port PB0 drive capability configuration register: 0: High drive capability; 1: Low drive capability
7	RSV	-	-	Reserved
6	PA6_DR	RW	0x0	Port PA6 drive capability configuration register: 0: High drive capability; 1: Low drive capability
5	PA5_DR	RW	0x0	Port PA5 drive capability configuration register: 0: High drive capability; 1: Low drive capability
4	PA4_DR	RW	0x0	Port PA4 drive capability configuration register: 0: High drive capability; 1: Low drive capability
3	PA3_DR	RW	0x0	Port PA3 drive capability configuration register: 0: High drive capability; 1: Low drive capability

Bit	Name	Attribute	Reset Value	Description
2	PA2_DR	RW	0x0	Port PA2 drive capability configuration register: 0: High drive capability; 1: Low drive capability
1	PA1_DR	RW	0x0	Port PA1 drive capability configuration register: 0: High drive capability; 1: Low drive capability
0	PA0_DR	RW	0x0	Port PA0 drive capability configuration register: 0: High drive capability; 1: Low drive capability

5.5.14 Port Pull-up Configuration Register SCU_PAD_PU (Offset: 06CH)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	PD3_PU	RW	0x0	Port PD3 pull-up configuration register: 0: Disabled; 1: Enabled
26	PD2_PU	RW	0x0	Port PD2 pull-up configuration register: 0: Disabled; 1: Enabled
25	PD1_PU	RW	0x0	Port PD1 pull-up configuration register: 0: Disabled; 1: Enabled
24	PD0_PU	RW	0x0	Port PD0 pull-up configuration register: 0: Disabled; 1: Enabled
23	RSV	-	-	Reserved
22	PC6_PU	RW	0x0	Port PC6 pull-up configuration register: 0: Disabled; 1: Enabled
21	PC5_PU	RW	0x1	Port PC5 pull-up configuration register: 0: Disabled; 1: Enabled
20	PC4_PU	RW	0x0	Port PC4 pull-up configuration register: 0: Disabled; 1: Enabled
19	PC3_PU	RW	0x0	Port PC3 pull-up configuration register: 0: Disabled; 1: Enabled
18	PC2_PU	RW	0x0	Port PC2 pull-up configuration register: 0: Disabled; 1: Enabled
17	PC1_PU	RW	0x0	Port PC1 pull-up configuration register: 0: Disabled; 1: Enabled

Bit	Name	Attribute	Reset Value	Description
16	PC0_PU	RW	0x0	Port PC0 pull-up configuration register: 0: Disabled; 1: Enabled
15	PB7_PU	RW	0x0	Port PB7 pull-up configuration register: 0: Disabled; 1: Enabled
14	PB6_PU	RW	0x0	Port PB6 pull-up configuration register: 0: Disabled; 1: Enabled
13	PB5_PU	RW	0x0	Port PB5 pull-up configuration register: 0: Disabled; 1: Enabled
12	PB4_PU	RW	0x0	Port PB4 pull-up configuration register: 0: Disabled; 1: Enabled
11	PB3_PU	RW	0x0	Port PB3 pull-up configuration register: 0: Disabled; 1: Enabled
10	PB2_PU	RW	0x0	Port PB2 pull-up configuration register: 0: Disabled; 1: Enabled
9	PB1_PU	RW	0x0	Port PB1 pull-up configuration register: 0: Disabled; 1: Enabled
8	PB0_PU	RW	0x0	Port PB0 pull-up configuration register: 0: Disabled; 1: Enabled
7	RSV	-	-	Reserved
6	PA6_PU	RW	0x0	Port PA6 pull-up configuration register: 0: Disabled; 1: Enabled
5	PA5_PU	RW	0x0	Port PA5 pull-up configuration register: 0: Disabled; 1: Enabled
4	PA4_PU	RW	0x0	Port PA4 pull-up configuration register: 0: Disabled; 1: Enabled
3	PA3_PU	RW	0x0	Port PA3 pull-up configuration register: 0: Disabled; 1: Enabled
2	PA2_PU	RW	0x1	Port PA2 pull-up configuration register: 0: Disabled; 1: Enabled
1	PA1_PU	RW	0x0	Port PA1 pull-up configuration register: 0: Disabled; 1: Enabled
0	PA0_PU	RW	0x0	Port PA0 pull-up configuration register: 0: Disabled; 1: Enabled

5.5.15 Port Pull-down Configuration Register SCU_PAD_PD (Offset: 078H)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	PD3_PD	RW	0x0	Port PD3 pull-down configuration register: 0: Disabled; 1: Enabled
26	PD2_PD	RW	0x0	Port PD2 pull-down configuration register: 0: Disabled; 1: Enabled
25	PD1_PD	RW	0x0	Port PD1 pull-down configuration register: 0: Disabled; 1: Enabled
24	PD0_PD	RW	0x0	Port PD0 pull-down configuration register: 0: Disabled; 1: Enabled
23	RSV	-	-	Reserved
22	PC6_PD	RW	0x1	Port PC6 pull-down configuration register: 0: Disabled; 1: Enabled
21	PC5_PD	RW	0x0	Port PC5 pull-down configuration register: 0: Disabled; 1: Enabled
20	PC4_PD	RW	0x0	Port PC4 pull-down configuration register: 0: Disabled; 1: Enabled
19	PC3_PD	RW	0x0	Port PC3 pull-down configuration register: 0: Disabled; 1: Enabled
18	PC2_PD	RW	0x0	Port PC2 pull-down configuration register: 0: Disabled; 1: Enabled
17	PC1_PD	RW	0x0	Port PC1 pull-down configuration register: 0: Disabled; 1: Enabled
16	PC0_PD	RW	0x0	Port PC0 pull-down configuration register: 0: Disabled; 1: Enabled
15	PB7_PD	RW	0x0	Port PB7 pull-down configuration register: 0: Disabled; 1: Enabled
14	PB6_PD	RW	0x0	Port PB6 pull-down configuration register: 0: Disabled; 1: Enabled
13	PB5_PD	RW	0x0	Port PB5 pull-down configuration register: 0: Disabled; 1: Enabled
12	PB4_PD	RW	0x0	Port PB4 pull-down configuration register: 0: Disabled; 1: Enabled

Bit	Name	Attribute	Reset Value	Description
11	PB3_PD	RW	0x0	Port PB3 pull-down configuration register: 0: Disabled; 1: Enabled
10	PB2_PD	RW	0x0	Port PB2 pull-down configuration register: 0: Disabled; 1: Enabled
9	PB1_PD	RW	0x0	Port PB1 pull-down configuration register: 0: Disabled; 1: Enabled
8	PB0_PD	RW	0x0	Port PB0 pull-down configuration register: 0: Disabled; 1: Enabled
7	RSV	-	-	Reserved
6	PA6_PD	RW	0x0	Port PA6 pull-down configuration register: 0: Disabled; 1: Enabled
5	PA5_PD	RW	0x0	Port PA5 pull-down configuration register: 0: Disabled; 1: Enabled
4	PA4_PD	RW	0x0	Port PA4 pull-down configuration register: 0: Disabled; 1: Enabled
3	PA3_PD	RW	0x0	Port PA3 pull-down configuration register: 0: Disabled; 1: Enabled
2	PA2_PD	RW	0x0	Port PA2 pull-down configuration register: 0: Disabled; 1: Enabled
1	PA1_PD	RW	0x0	Port PA1 pull-down configuration register: 0: Disabled; 1: Enabled
0	PA0_PD	RW	0x0	Port PA0 pull-down configuration register: 0: Disabled; 1: Enabled

5.5.16 Port Open-drain Output Configuration Register SCU_PAD_OD (Offset: 084H)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	PD3_OD	RW	0x0	Port PD3 open-drain output configuration register: 0: Disabled; 1: Enabled
26	PD2_OD	RW	0x0	Port PD2 open-drain output configuration register: 0: Disabled; 1: Enabled

Bit	Name	Attribute	Reset Value	Description
25	PD1_OD	RW	0x0	Port PD1 open-drain output configuration register: 0: Disabled; 1: Enabled
24	PD0_OD	RW	0x0	Port PD0 open-drain output configuration register: 0: Disabled; 1: Enabled
23	RSV	-	-	Reserved
22	PC6_OD	RW	0x0	Port PC6 open-drain output configuration register: 0: Disabled; 1: Enabled
21	PC5_OD	RW	0x0	Port PC5 open-drain output configuration register: 0: Disabled; 1: Enabled
20	PC4_OD	RW	0x0	Port PC4 open-drain output configuration register: 0: Disabled; 1: Enabled
19	PC3_OD	RW	0x0	Port PC3 open-drain output configuration register: 0: Disabled; 1: Enabled
18	PC2_OD	RW	0x0	Port PC2 open-drain output configuration register: 0: Disabled; 1: Enabled
17	PC1_OD	RW	0x0	Port PC1 open-drain output configuration register: 0: Disabled; 1: Enabled
16	PC0_OD	RW	0x0	Port PC0 open-drain output configuration register: 0: Disabled; 1: Enabled
15	PB7_OD	RW	0x0	Port PB7 open-drain output configuration register: 0: Disabled; 1: Enabled
14	PB6_OD	RW	0x0	Port PB6 open-drain output configuration register: 0: Disabled; 1: Enabled
13	PB5_OD	RW	0x0	Port PB5 open-drain output configuration register: 0: Disabled; 1: Enabled
12	PB4_OD	RW	0x0	Port PB4 open-drain output configuration register: 0: Disabled; 1: Enabled
11	PB3_OD	RW	0x0	Port PB3 open-drain output configuration register: 0: Disabled; 1: Enabled
10	PB2_OD	RW	0x0	Port PB2 open-drain output configuration register: 0: Disabled; 1: Enabled
9	PB1_OD	RW	0x0	Port PB1 open-drain output configuration register: 0: Disabled; 1: Enabled
8	PB0_OD	RW	0x0	Port PB0 open-drain output configuration register: 0: Disabled; 1: Enabled
7	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
6	PA6_OD	RW	0x0	Port PA6 open-drain output configuration register: 0: Disabled; 1: Enabled
5	PA5_OD	RW	0x0	Port PA5 open-drain output configuration register: 0: Disabled; 1: Enabled
4	PA4_OD	RW	0x0	Port PA4 open-drain output configuration register: 0: Disabled; 1: Enabled
3	PA3_OD	RW	0x0	Port PA3 open-drain output configuration register: 0: Disabled; 1: Enabled
2	PA2_OD	RW	0x0	Port PA2 open-drain output configuration register: 0: Disabled; 1: Enabled
1	PA1_OD	RW	0x0	Port PA1 open-drain output configuration register: 0: Disabled; 1: Enabled
0	PA0_OD	RW	0x0	Port PA0 open-drain output configuration register: 0: Disabled; 1: Enabled

5.5.17 Port Input Type Configuration Register SCU_PAD_CS (Offset: 090H)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	PD3_CS	RW	0x0	Port PD3 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
26	PD2_CS	RW	0x0	Port PD2 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
25	PD1_CS	RW	0x0	Port PD1 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
24	PD0_CS	RW	0x0	Port PD0 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
23	RSV	-	-	Reserved
22	PC6_CS	RW	0x0	Port PC6 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
21	PC5_CS	RW	0x0	Port PC5 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer

Bit	Name	Attribute	Reset Value	Description
20	PC4_CS	RW	0x0	Port PC4 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
19	PC3_CS	RW	0x0	Port PC3 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
18	PC2_CS	RW	0x0	Port PC2 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
17	PC1_CS	RW	0x0	Port PC1 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
16	PC0_CS	RW	0x0	Port PC0 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
15	PB7_CS	RW	0x0	Port PB7 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
14	PB6_CS	RW	0x0	Port PB6 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
13	PB5_CS	RW	0x0	Port PB5 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
12	PB4_CS	RW	0x0	Port PB4 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
11	PB3_CS	RW	0x0	Port PB3 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
10	PB2_CS	RW	0x0	Port PB2 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
9	PB1_CS	RW	0x0	Port PB1 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
8	PB0_CS	RW	0x0	Port PB0 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
7	RSV	-	-	Reserved
6	PA6_CS	RW	0x0	Port PA6 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
5	PA5_CS	RW	0x0	Port PA5 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
4	PA4_CS	RW	0x0	Port PA4 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
3	PA3_CS	RW	0x0	Port PA3 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer
2	PA2_CS	RW	0x0	Port PA2 input type configuration register: 0: Schmitt input buffer; 1: CMOS input buffer

Bit	Name	Attribute	Reset Value	Description
1	PA1_CS	RW	0x0	Port PA1 input type configuration register 0: Schmitt input buffer; 1: CMOS input buffer
0	PA0_CS	RW	0x0	Port PA0 input type configuration register 0: Schmitt input buffer; 1: CMOS input buffer

5.5.18 Port Input Enable Register SCU_PAD_IE (Offset: 09CH)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	PD3_IE	RW	0x0	Port PD3 input enable register: 0: Disabled; 1: Enabled
26	PD2_IE	RW	0x0	Port PD2 input enable register: 0: Disabled; 1: Enabled
25	PD1_IE	RW	0x0	Port PD1 input enable register: 0: Disabled; 1: Enabled
24	PD0_IE	RW	0x0	Port PD0 input enable register: 0: Disabled; 1: Enabled
23	RSV	-	-	Reserved
22	PC6_IE	RW	0x1	Port PC6 input enable register: 0: Disabled; 1: Enabled
21	PC5_IE	RW	0x1	Port PC5 input enable register: 0: Disabled; 1: Enabled
20	PC4_IE	RW	0x0	Port PC4 input enable register: 0: Disabled; 1: Enabled
19	PC3_IE	RW	0x0	Port PC3 input enable register: 0: Disabled; 1: Enabled
18	PC2_IE	RW	0x0	Port PC2 input enable register: 0: Disabled; 1: Enabled
17	PC1_IE	RW	0x0	Port PC1 input enable register: 0: Disabled; 1: Enabled
16	PC0_IE	RW	0x0	Port PC0 input enable register: 0: Disabled; 1: Enabled
15	PB7_IE	RW	0x0	Port PB7 input enable register: 0: Disabled; 1: Enabled
14	PB6_IE	RW	0x0	Port PB6 input enable register: 0: Disabled; 1: Enabled

Bit	Name	Attribute	Reset Value	Description
13	PB5_IE	RW	0x0	Port PB5 input enable register: 0: Disabled; 1: Enabled
12	PB4_IE	RW	0x0	Port PB4 input enable register: 0: Disabled; 1: Enabled
11	PB3_IE	RW	0x0	Port PB3 input enable register: 0: Disabled; 1: Enabled
10	PB2_IE	RW	0x0	Port PB2 input enable register: 0: Disabled; 1: Enabled
9	PB1_IE	RW	0x0	Port PB1 input enable register: 0: Disabled; 1: Enabled
8	PB0_IE	RW	0x0	Port PB0 input enable register: 0: Disabled; 1: Enabled
7	RSV	-	-	Reserved
6	PA6_IE	RW	0x0	Port PA6 input enable register: 0: Disabled; 1: Enabled
5	PA5_IE	RW	0x0	Port PA5 input enable register: 0: Disabled; 1: Enabled
4	PA4_IE	RW	0x0	Port PA4 input enable register: 0: Disabled; 1: Enabled
3	PA3_IE	RW	0x0	Port PA3 input enable register: 0: Disabled; 1: Enabled
2	PA2_IE	RW	0x1	Port PA2 input enable register: 0: Disabled; 1: Enabled
1	PA1_IE	RW	0x0	Port PA1 input enable register: 0: Disabled; 1: Enabled
0	PA0_IE	RW	0x0	Port PA0 input enable register: 0: Disabled; 1: Enabled

5.5.19 Port Input Level Register SCU_PAD_STATUS (Offset: 0A4H)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	PD3_I	R	0x0	Port PD3 input level register: 0: Input low level; 1: Input high level
26	PD2_I	R	0x0	Port PD2 input level register: 0: Input low level; 1: Input high level

Bit	Name	Attribute	Reset Value	Description
25	PD1_I	R	0x0	Port PD1 input level register: 0: Input low level; 1: Input high level
24	PD0_I	R	0x0	Port PD0 input level register: 0: Input low level; 1: Input high level
23	RSV	-	-	Reserved
22	PC6_I	R	0x1	Port PC6 input level register: 0: Input low level; 1: Input high level
21	PC5_I	R	0x1	Port PC5 input level register: 0: Input low level; 1: Input high level
20	PC4_I	R	0x0	Port PC4 input level register: 0: Input low level; 1: Input high level
19	PC3_I	R	0x0	Port PC3 input level register: 0: Input low level; 1: Input high level
18	PC2_I	R	0x0	Port PC2 input level register: 0: Input low level; 1: Input high level
17	PC1_I	R	0x0	Port PC1 input level register: 0: Input low level; 1: Input high level
16	PC0_I	R	0x0	Port PC0 input level register: 0: Input low level; 1: Input high level
15	PB7_I	R	0x0	Port PB7 input level register: 0: Input low level; 1: Input high level
14	PB6_I	R	0x0	Port PB6 input level register: 0: Input low level; 1: Input high level
13	PB5_I	R	0x0	Port PB5 input level register: 0: Input low level; 1: Input high level
12	PB4_I	R	0x0	Port PB4 input level register: 0: Input low level; 1: Input high level
11	PB3_I	R	0x0	Port PB3 input level register: 0: Input low level; 1: Input high level
10	PB2_I	R	0x0	Port PB2 input level register: 0: Input low level; 1: Input high level
9	PB1_I	R	0x0	Port PB1 input level register: 0: Input low level; 1: Input high level
8	PB0_I	R	0x0	Port PB0 input level register: 0: Input low level; 1: Input high level
7	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
6	PA6_I	R	0x0	Port PA6 input level register: 0: Input low level; 1: Input high level
5	PA5_I	R	0x0	Port PA5 input level register: 0: Input low level; 1: Input high level
4	PA4_I	R	0x0	Port PA4 input level register: 0: Input low level; 1: Input high level
3	PA3_I	R	0x0	Port PA3 input level register: 0: Input low level; 1: Input high level
2	PA2_I	R	0x1	Port PA2 input level register: 0: Input low level; 1: Input high level
1	PA1_I	R	0x0	Port PA1 input level register: 0: Input low level; 1: Input high level
0	PA0_I	R	0x0	Port PA0 input level register: 0: Input low level; 1: Input high level

5.5.20 Port Speed Configuration Register SCU_PAD_SR (Offset: 0A8H)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	PD3_SR	RW	0x1	Port PD3 speed configuration register: 0: High speed; 1: Low speed
26	PD2_SR	RW	0x1	Port PD2 speed configuration register: 0: High speed; 1: Low speed
25	PD1_SR	RW	0x1	Port PD1 speed configuration register: 0: High speed; 1: Low speed
24	PD0_SR	RW	0x1	Port PD0 speed configuration register: 0: High speed; 1: Low speed
23	RSV	-	-	Reserved
22	PC6_SR	RW	0x1	Port PC6 speed configuration register: 0: High speed; 1: Low speed
21	PC5_SR	RW	0x1	Port PC5 speed configuration register: 0: High speed; 1: Low speed
20	PC4_SR	RW	0x1	Port PC4 speed configuration register: 0: High speed; 1: Low speed

Bit	Name	Attribute	Reset Value	Description
19	PC3_SR	RW	0x1	Port PC3 speed configuration register: 0: High speed; 1: Low speed
18	PC2_SR	RW	0x1	Port PC2 speed configuration register: 0: High speed; 1: Low speed
17	PC1_SR	RW	0x1	Port PC1 speed configuration register: 0: High speed; 1: Low speed
16	PC0_SR	RW	0x1	Port PC0 speed configuration register: 0: High speed; 1: Low speed
15	PB7_SR	RW	0x1	Port PB7 speed configuration register: 0: High speed; 1: Low speed
14	PB6_SR	RW	0x1	Port PB6 speed configuration register: 0: High speed; 1: Low speed
13	PB5_SR	RW	0x1	Port PB5 speed configuration register: 0: High speed; 1: Low speed
12	PB4_SR	RW	0x1	Port PB4 speed configuration register: 0: High speed; 1: Low speed
11	PB3_SR	RW	0x1	Port PB3 speed configuration register: 0: High speed; 1: Low speed
10	PB2_SR	RW	0x1	Port PB2 speed configuration register: 0: High speed; 1: Low speed
9	PB1_SR	RW	0x1	Port PB1 speed configuration register: 0: High speed; 1: Low speed
8	PB0_SR	RW	0x1	Port PB0 speed configuration register: 0: High speed; 1: Low speed
7	RSV	-	-	Reserved
6	PA6_SR	RW	0x1	Port PA6 speed configuration register: 0: High speed; 1: Low speed
5	PA5_SR	RW	0x1	Port PA5 speed configuration register: 0: High speed; 1: Low speed
4	PA4_SR	RW	0x1	Port PA4 speed configuration register: 0: High speed; 1: Low speed
3	PA3_SR	RW	0x1	Port PA3 speed configuration register: 0: High speed; 1: Low speed
2	PA2_SR	RW	0x1	Port PA2 speed configuration register: 0: High speed; 1: Low speed
1	PA1_SR	RW	0x1	Port PA1 speed configuration register: 0: High speed; 1: Low speed

Bit	Name	Attribute	Reset Value	Description
0	PA0_SR	RW	0x1	Port PA0 speed configuration register: 0: High speed; 1: Low speed

5.5.21 IO Control Protection Register SCU_IOCTRL_PROT (Offset: 0B4H)

Bit	Name	Attribute	Reset Value	Description
31:0	IOCTRL_PROTECT	RW	0x0	Protection control register for IO registers PA_SEL/ PB_SEL/ PC_SEL/ PD_SEL/ PAD_ADS/ PAD_DR/ PAD_PU/ PAD_PD/ PAD_OD/ PAD_CS/ PAD_IE/ PAD_SR. Writing 0xA5A5_5A5A to this register enables writing to the specified IO registers. After configuring the IO register, the write access thereof will not be automatically disabled. Writing any other value disables the write access to these IO registers. Reading the register returns the write enable status of the IO registers: 0: Write not enabled 1: Write enabled

5.5.22 LVD Configuration Register SCU_LVD_CFG (Offset: 0B8H)

Bit	Name	Attribute	Reset Value	Description
31	BUF_VADJ	RW	0x0	BUF output voltage setting: 1: BUF outputs 1.5 V 0: BUF outputs 0.8 V
30	BUF_O2P	RW	0x0	BUF output to PD2 enable: 1: BUF outputs to PD2 0: BUF does not output to PD2
29	BUF_EN	RW	0x0	BUF enable: 1: BUF enabled 0: BUF disabled

<table>
<tr><th>Bit</th><th>Name</th><th>Attribute</th><th>Reset Value</th><th>Description</th></tr>
<tr><td>28:25</td><td>RSV</td><td>-</td><td>-</td><td>Reserved</td></tr>
<tr><td>24</td><td>LVD_LVEN</td><td>RW</td><td>0x1</td><td>LVD filter enable bit</td></tr>
<tr><td>23:16</td><td>LVD_FILTER</td><td>RW</td><td>0x20</td><td>LVD filter configuration bit:
0: Filter out 1 glitch of the 32K low-speed system clock for LVD
1: Filter out 2 glitches of the 32K low-speed system clock for LVD
……
255: Filter out 256 glitches of the 32K low-speed system clock for LVD</td></tr>
<tr><td>15:14</td><td>RSV</td><td>-</td><td>-</td><td>Reserved</td></tr>
<tr><td>13:10</td><td>LVRS</td><td>RW</td><td>0x0</td><td>LVR point voltage setting:
<table>
<tr><th>LVRS</th><th>LVR POINT</th><th>LVRS</th><th>LVR POINT</th></tr>
<tr><td>0000</td><td>1.89</td><td>1000</td><td>4.2</td></tr>
<tr><td>0001</td><td>2.1</td><td>1001</td><td>4.48</td></tr>
<tr><td>0010</td><td>2.39</td><td>1010</td><td>4.81</td></tr>
<tr><td>0011</td><td>2.7</td><td>1011</td><td>Reserved</td></tr>
<tr><td>0100</td><td>3</td><td>1100</td><td>Reserved</td></tr>
<tr><td>0101</td><td>3.3</td><td>1101</td><td>Reserved</td></tr>
<tr><td>0110</td><td>3.6</td><td>1110</td><td>Reserved</td></tr>
<tr><td>0111</td><td>3.87</td><td>1111</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>9</td><td>LVD_INTR_EN</td><td>RW</td><td>0x0</td><td>LVD interrupt enable bit:
0: LVD interrupt disabled
1: LVD interrupt enabled</td></tr>
<tr><td>8</td><td>LVD_RESET_EN</td><td>RW</td><td>0x0</td><td>LVD reset enable bit:
0: LVD reset disabled
1: LVD reset enabled</td></tr>
<tr><td>7:4</td><td>LVDS</td><td>RW</td><td>0x0</td><td>LVD point voltage setting:
<table>
<tr><th>LVDS</th><th>LVD POINT</th><th>LVDS</th><th>LVD POINT</th></tr>
<tr><td>0000</td><td>1.89</td><td>1000</td><td>4.2</td></tr>
<tr><td>0001</td><td>2.1</td><td>1001</td><td>4.48</td></tr>
<tr><td>0010</td><td>2.39</td><td>1010</td><td>4.81</td></tr>
<tr><td>0011</td><td>2.7</td><td>1011</td><td>Reserved</td></tr>
<tr><td>0100</td><td>3</td><td>1100</td><td>Reserved</td></tr>
<tr><td>0101</td><td>3.3</td><td>1101</td><td>Reserved</td></tr>
</table></td></tr>
</table>

Bit	Name	Attribute	Reset Value	Description
				0110 3.6 1110 Reserved 0111 3.87 1111 Reserved
3:1	RSV	-	-	Reserved
0	LVD_EN	RW	0x0	LVD module enable register: 0: Disabled 1: Enabled

5.5.23 PGA Control Register SCU_PGA_CFG (Offset: 0BCH)

Bit	Name	Attribute	Reset Value	Description
31:26	RSV	-	-	Reserved
25	PGA_VREFSEL	RW	0x0	VREF voltage selection signal: 0: VREFLDO voltage, 1.2 V/3.6 V/4.2 V/AVDD 1: Voltage input on PD3
24	PGA1_VSEL	RW	0x0	PGA1 and PGA2 output common-mode voltage selection signal: 0: VREF/2 (voltage selected by PGA_VREFSEL) 1: VBG (BUF LDO)
23	PGA0_VSEL	RW	0x0	PGA0 output common-mode voltage selection signal: 0: VREF/2 (voltage selected by PGA_VREFSEL) 1: VBG (BUF LDO)
22:21	PGA0_RSEL	RW	0x0	VOUT0 output switch on-resistance selection signal: 00: 200 Ω 01: 1 kΩ 10: 10 kΩ 11: HiZ
20	PGA2_POSSEL	RW	0x0	PGA2 positive input voltage selection signal: 0: External input, PC4 input 1: Internally connected to AVSS
19	PGA1_POSSEL	RW	0x0	PGA1 positive input voltage selection signal: 0: External input, PC1 input 1: Internally connected to AVSS

Bit	Name	Attribute	Reset Value	Description
18	PGA0_POSSEL	RW	0x0	PGA0 positive input voltage selection signal: 0: External input, PB6 input 1: Internally connected to AVSS
17	PGA2_NEGSEL	RW	0x0	PGA2 negative input voltage selection signal: 0: External input, PC3 input 1: Internally connected to AVSS
16	PGA1_NEGSEL	RW	0x0	PGA1 negative input voltage selection signal: 0: External input, PC0 input 1: Internally connected to AVSS
15	PGA0_NEGSEL	RW	0x0	PGA0 negative input voltage selection signal: 0: External input, PB5 input 1: Internally connected to AVSS
14:12	PGA2_SELGAIN	RW	0x0	PGA2 gain control signal: PGA2_SELGAIN = 000, 1x gain PGA2_SELGAIN = 001, 2x gain PGA2_SELGAIN = 010, 4x gain PGA2_SELGAIN = 011, 8x gain PGA2_SELGAIN = 100, 16x gain PGA2_SELGAIN = 101, 32x gain PGA2_SELGAIN = 110, 64x gain
11:9	PGA1_SELGAIN	RW	0x0	PGA1 gain control signal: PGA1_SELGAIN = 000, 1x gain PGA1_SELGAIN = 001, 2x gain PGA1_SELGAIN = 010, 4x gain PGA1_SELGAIN = 011, 8x gain PGA1_SELGAIN = 100, 16x gain PGA1_SELGAIN = 101, 32x gain PGA1_SELGAIN = 110, 64x gain
8:6	PGA0_SELGAIN	RW	0x0	PGA0 gain control signal: PGA0_SELGAIN = 000, 1x gain PGA0_SELGAIN = 001, 2x gain PGA0_SELGAIN = 010, 4x gain PGA0_SELGAIN = 011, 8x gain PGA0_SELGAIN = 100, 16x gain PGA0_SELGAIN = 101, 32x gain PGA0_SELGAIN = 110, 64x gain

Bit	Name	Attribute	Reset Value	Description
5	PGA2_EN	RW	0x0	Enable signal, PGA2 works when EN_PGA2 = 1
4	PGA1_EN	RW	0x0	Enable signal, PGA1 works when EN_PGA1 = 1
3	PGA0_EN	RW	0x0	Enable signal, PGA0 works when EN_PGA0 = 1
2	PGA2_EN2P	RW	0x0	VOUT2 output switch enable signal: 0: VOUT2PAD = HiZ 1: VOUT2PAD = VOUT2
1	PGA1_EN2P	RW	0x0	VOUT1 output switch enable signal: 0: VOUT1PAD = HiZ 1: VOUT1PAD = VOUT1
0	PGA0_EN2P	RW	0x0	VOUT0 output switch enable signal: 0: VOUT0PAD = HiZ 1: VOUT0PAD = VOUT0

5.5.24 External Reset Port Selection Register SCU_EXTRST_SEL (Offset: 0D0H)

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	RESETN_SEL	RW	0x0	External reset port selection register. This bit is writable only when the high 16 bits [31:16] of this register are written with 0xA5A5. 1: The external reset signal is invalid. This pin can be used for other functions of PA2. 0: The external reset signal is valid.

5.5.25 Stop Mode Selection Register SCU_STOP_SEL (Offset: 0D4H)

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	STOPMODE_SEL	RW	0x0	Stop mode selection register. This bit is writable only when the high 16 bits [31:16] of this register are written with 0xA5A5.

Bit	Name	Attribute	Reset Value	Description
				1: Stop mode is valid. 0: Stop mode is invalid.

5.5.26 Software Reset Register SCU_SOFT_RSTN (Offset: 0D8H)

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	GRSTN	W	0x0	Writing 0 to this register will trigger a reset of the chip.

5.5.27 Interrupt Vector Address Remapping Register SCU_VET_OFFSET (Offset: 0DCH)

Bit	Name	Attribute	Reset Value	Description
31:10	VECTOROFFSET	RW	0x0	After enabling the interrupt vector remapping function, the base address of the interrupt vector is {VECTOROFFSET, 10{1'b0}}.
9:1	RSV	-	-	Reserved
0	VECTOROFFSET_EN	RW	0x0	Interrupt vector remapping function enable: 1: Disabled 0: Enabled

5.5.28 LVD Interrupt Register SCU_LVD_INTR (Offset: FCH)

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	LVD_INTR	W1C	0x0	LVD interrupt register: 1: LVD generates an interrupt. 0: LVD does not generate an interrupt. Software writing 1 clears this bit.

5.5.29 VREF Control Register SCU_VREF (Offset: 188H)

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved
2:1	VREF_SEL	RW	0x0	VREF selection: 00: 1.2 V 01: 3.6 V 10: 4.2 V 11: Analog supply voltage
0	VREF_EN	RW	0x0	VREF enable

6 EFC

6.1 Overview

The chip integrates a 32KB eFlash memory for storing all the key offline information and data of the chip. EFC is an eFlash controller, which performs Flash reading, writing, erasing and other operations with the cooperation of CPU.

6.2 Main Features

- Supporting eFlash read/write (8/16/32 bits), sector erase and chip erase
- Configurable read wait time
- 64 sectors in the main area, each with 512 bytes
- Write protection
- Automatic bus locking
- Supporting multi-level protection mechanisms to ensure the security, integrity and reliability of eFlash data

6.3 eFlash Data Protection Mechanisms

- SWD hardware protection

  By configuring the keyword “SWD_DISABLE_PIN” in the EEPROM, the SWD interface is hardware-disabled, prohibiting communication through the SWD interface, thereby preventing physical access or illegal debugging via the SWD interface, safeguarding against malicious code injection or firmware tampering.

- SWD read protection

  By configuring the keyword “SWD_READ_DISABLE” in the EEPROM, read operations on the eFlash memory through the SWD interface are prohibited, protecting firmware code and sensitive data from being extracted. Customers can only unlock read operations on the eFlash in this power-on cycle by erasing the main area of the eFlash.

- eFlash sector write protection

  The 32KB main area of the eFlash memory is managed in partitions, and write protection for data in different areas is achieved by configuring the keyword “OTP_MAIN_EN” in the EEPROM.

- ECC function

  Error-correcting code technology is used to detect and correct read data errors in eFlash memory cells, ensuring data integrity and preventing read data corruption caused by hardware aging or environmental interference.

- EEPROM sector write protection

  The EEPROM storage space is managed in partitions, and erase/write protection for data in different areas is achieved by configuring the keywords “OTP_EEP0_EN”, “OTP_EEP1_EN” and “OTP_EEP2_EN” in the OTP. Once the OTP keywords are enabled, the values in the corresponding EEPROM areas cannot be rewritten, including keywords such as SWD_READ_DISABLE, SWD_DISABLE_PIN, and OTP_MAIN_EN.

6.3.1 eFlash Storage Area

The eFlash storage space is mainly divided into the Main area, EEPROM area, and NVR area. The Main area is 32KB in size and is primarily used for storing application programs and application data. The EEPROM area is used for storing some application data. Erase operations

on the Main area do not affect the data stored in the EEPROM area. The NVR area is used for storing configuration information that is set at the factory, the internal values stored in this area cannot be erased or rewritten after the chip leaves the factory.

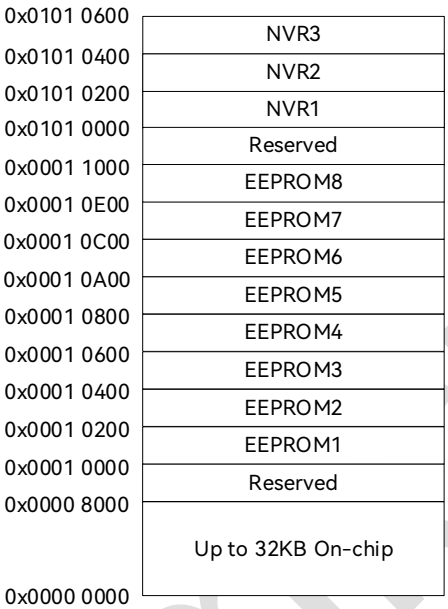


Figure 6-1: eFlash Storage Area

6.3.2 EEPROM Configuration Words

The figure below shows the keyword configuration in EEPROM8.

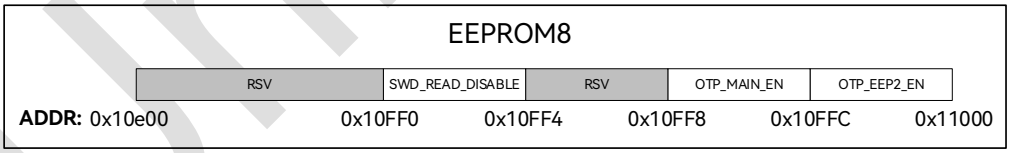


Figure 6-2: EEPROM8 Keyword Configuration Diagram

When the value of SWD_READ_DISABLE is set to 0x55AAAA55, the SWD read protection function is activated after the chip is externally reset or powered on again. Once this function is activated, any attempt to read data from eFlash via SWD will return 0xFFFFFFFF, regardless of the actual data stored in eFlash. This function can only be disabled by performing a ChipErase operation on the eFlash Main area during the current external reset release or power-on period. If the value of SWD_READ_DISABLE remains 0x55AAAA55, the SWD read protection function

will be reactivated after an external reset or power cycle.

OTP_MAIN_EN is the enable bit for eFlash sector write protection. Each bit of OTP_MAIN_EN[15:0] represents whether a 2KB section of the Main area is subject to write protection. The 32KB Main area is divided into 16 sections, each controlled by a bit in OTP_MAIN_EN[15:0] to determine whether software erase/write operations are allowed. A value of 1 for the corresponding bit means that the corresponding section is protected from erase/write operations, while a value of 0 means that erase/write operations are allowed. OTP_MAIN_EN[0] controls the area from address 0x0 to 0x800, OTP_MAIN_EN[1] controls the area from address 0x800 to 0x1000, and so on, with OTP_MAIN_EN[15] controlling the area from address 0x7800 to 0x8000. The keyword stored in OTP_MAIN_EN[15:0] only takes effect when it is the bitwise inverse of OTP_MAIN_EN[31:16]. Otherwise, regardless of whether the bits in OTP_MAIN_EN[15:0] are 0 or 1, they will not affect the erase/write operations on the eFlash Main area.

OTP_EEP2_EN is the enable bit for EEPROM8 erase/write protection. When its value is 0x55AAAA55, the chip will enable EEPROM8 erase/write protection after an external reset or power cycle. This means that the EEPROM8 area, including the keywords SWD_READ_DISABLE, OTP_MAIN_EN, and OTP_EEP2_EN, will be protected from erase/write operations.

6.3.3 SWD Hardware Disable

The keyword for SWD hardware disable, SWD_DISABLE_PIN, is located in NVR0 and needs to be enabled by calling the existing library functions provided in the chip SDK. Once this function is enabled, the SWD-related IOs of the chip will no longer be connected to the CPU interface SWD signals, effectively disabling the SWD interface at the hardware level.

6.4 Register Description

Register base address: 0x0110_0000

Table 6-1: List of EFC Registers

Offset	Name	Description
0x00	EFC_CTRL	EFC control register
0x04	EFC_OTPSTATUS	EFC OTP status register
0x08	EFC_KEY	EFC KEY input register
0x0c	EFC_INTSTATUS	EFC interrupt status register
0x10	EFC_INTEN	EFC interrupt enable register
0x14	EFC_EP_CTRL	EFC erase/write control register
0x1C	EFC_LDOTRIM	EFC LDO TRIM register
0x20	EFC_RCHTRIM	EFC RCH TRIM register
0x24	EFC_RCLTRIM	EFC RCL TRIM register
0x28	EFC_PGA0TRIM	EFC PGA0 TRIM register
0x2C	EFC_PGA1TRIM	EFC PGA1 TRIM register
0x30	EFC_ADCTRIM	EFC ADC TRIM register

6.4.1 EFC Control Register (EFC_CTRL) (Offset: 00H)

Bit	Name	Attribute	Default Value	Functional Description
31:21	RSV	-	-	Reserved
20	VMON3	RW	0x0	VMON3 mode setting bit: 1: VMON3 mode enabled 0: VMON3 mode disabled For test use only, not available to customers.
19	VMON2	RW	0x0	VMON2 mode setting bit: 1: VMON2 mode enabled 0: VMON2 mode disabled For test use only, not available to customers.
18	VMON1	RW	0x0	VMON1 mode setting bit: 1: VMON1 mode enabled 0: VMON1 mode disabled For test use only, not available to customers.

Bit	Name	Attribute	Default Value	Functional Description
17	READM1	RW	0x0	READM1 mode setting bit: 1: READM1 mode enabled 0: READM1 mode disabled For test use only, not available to customers.
16	READM0	RW	0x0	READM0 mode setting bit: 1: READM0 mode enabled 0: READM0 mode disabled For test use only, not available to customers.
15:12	RSV	-	-	Reserved
11	PRFM_EN	RW	0x0	Prefetch mode with multiple (4) buffer caches: 1: Prefetch multiple-buffer mode enabled 1: Prefetch multiple-buffer mode disabled
10	PRFS_EN	RW	0x0	Prefetch mode with a single buffer cache: 1: Prefetch single-buffer mode enabled 1: Prefetch single-buffer mode disabled
9	KEYERR_CLR	RW	0x0	When erasing or writing Flash, if the input KEY does not match, the control register can be released from the ERROR state: 1: Release ERROR lock state 0: Maintain current state unchanged
8	PRF_EN	RW	0x0	Prefetch mode setting bit: 1: Prefetch mode enabled 1: Prefetch mode disabled
7:1	RSV	-	-	Reserved
0	RD_WAIT	RW	0x1	Flash read wait cycle configuration: 0: Wait for 0 system clock cycle 1: Wait for 1 system clock cycle

6.4.2 EFC OTP Status Register EFC_OTPSTATUS (Offset: 04H)

Bit	Name	Attribute	Default Value	Functional Description
31:16	OTP_MAIN	R	0x0	Main area OTP erase/write lock status bit. Each bit represents the erasable/writable flag of the Flash Main area, controlling a 2KB area:

Bit	Name	Attribute	Default Value	Functional Description
				OTP_MAIN[0]: If this bit is 1, the 0~2KB area is not allowed to be erased or written. … OTP_MAIN[15]: If this bit is 1, the 30~32KB area is not allowed to be erased or written.
15:9	RSV	-	-	Reserved
8	OTP_NVR2	R	0x0	NVR2 area OTP erase/write lock status bit: 1: NVR2 area cannot be erased or written. 0: NVR2 area can be erased or written.
7	OTP_NVR1	R	0x0	NVR1 area OTP erase/write lock status bit: 1: NVR1 area cannot be erased or written. 0: NVR1 area can be erased or written.
6	OTP_NVR0	R	0x0	NVR0 area OTP erase/write lock status bit: 1: NVR0 area cannot be erased or written. 0: NVR0 area can be erased or written.
5	OTP_EEP2	R	0x0	OTP erase/write lock status bit of the 8th sector of the EEPROM area: 1: The 8th sector of the EEPROM cannot be erased or written. 0: The 8th sector of the EEPROM can be erased or written.
4	OTP_EEP1	R	0x0	OTP erase/write lock status bit of the 7th sector of the EEPROM area: 1: The 7th sector of the EEPROM cannot be erased or written. 0: The 7th sector of the EEPROM can be erased or written.
3	OTP_EEP0	R	0x0	OTP erase/write lock status bit of the 6th sector of the EEPROM area: 1: The 6th sector of the EEPROM cannot be erased or written. 0: The 6th sector of the EEPROM can be erased or written.
2:0	RSV	-	-	Reserved

6.4.3 EFC KEY Input Register (EFC_KEY) (Offset: 08H)

Bit	Name	Attribute	Default Value	Functional Description
31:0	KEY	W	0x0	The Flash erase key input register must be correctly written with a valid KEY sequence at this address by software or SWD before initiating the erase or write operations. This is a dummy address with no physical register implemented.

6.4.4 EFC Interrupt Status Register (EFC_INTSTATUS) (Offset: 0CH)

Bit	Name	Attribute	Default Value	Functional Description
31:16	MAIN_ERR	W1C	0x0	Main area operation error. When the corresponding Main area is locked, performing write/erase operations on this Main area or a chip erase operation will set this bit to 1. MAIN_ERR[0]: If this bit is 1, an operation error has occurred in the 0–2KB area. … MAIN_ERR[0]: If this bit is 1, an operation error has occurred in the 30–32KB area. The corresponding bit can be cleared by software writing 1.
15:12	RSV	-	-	Reserved
11:9	KEY_STATE	R	0x0	3’b000: Default state 3’b001: Chip erase operation key is correct. 3’b010: Sector erase operation key is correct. 3’b011: Normal program operation key is correct. 3’b100: Key error
8	NVR2_ERR	W1C	0x0	1: An operation error has occurred in the NVR area. When the corresponding NVR2 area is locked, performing write/erase operations on this NVR2 area or a chip erase operation will set this bit to 1.

Bit	Name	Attribute	Default Value	Functional Description
				0: Normal state Software writing 1 clears this bit.
7	NVR1_ERR	W1C	0x0	1: An operation error has occurred in the NVR area. When the corresponding NVR1 area is locked, performing write/erase operations on this NVR1 area or a chip erase operation will set this bit to 1. 0: Normal state Software writing 1 clears this bit.
6	NVR0_ERR	W1C	0x0	1: An operation error has occurred in the NVR area. When the corresponding NVR0 area is locked, performing write/erase operations on this NVR0 area or a chip erase operation will set this bit to 1. 0: Normal state Software writing 1 clears this bit.
5	EEP2_ERR	W1C	0x0	1: An operation error has occurred in the EEPROM area. When the corresponding EEPROM2 area is locked, performing write/erase operations on this EEPROM2 area or a chip erase operation will set this bit to 1. 0: Normal state Software writing 1 clears this bit.
4	EEP1_ERR	W1C	0x0	1: An operation error has occurred in the EEPROM area. When the corresponding EEPROM1 area is locked, performing write/erase operations on this EEPROM1 area or a chip erase operation will set this bit to 1. 0: Normal state Software writing 1 clears this bit.
3	EEP0_ERR	W1C	0x0	1: An operation error has occurred in the EEPROM area. When the corresponding EEPROM0 area is locked, performing write/erase operations on this EEPROM0 area or a chip erase operation will set this bit to 1. 0: Normal state

Bit	Name	Attribute	Default Value	Functional Description
				Software writing 1 clears this bit.
2	KEY_ERR	W1C	0x0	Error status bit of key input during Flash erase/write 1: The input key is incorrect. 0: The input key is correct. Software writing 1 clears this bit.
1	PR_DONE	W1C	0x0	Write completion interrupt status bit: 1: Write completed; writing 1 clears this bit. If interrupts are enabled, an interrupt will be generated. 0: Write not completed
0	ER_DONE	W1C	0x0	Erase completion interrupt status bit: 1: Erase completed; writing 1 clears this bit. If interrupts are enabled, an interrupt will be generated. 0: Erase not completed

6.4.5 EFC Interrupt Enable Register EFC_INTEN (Offset: 10H)

Bit	Name	Attribute	Default Value	Functional Description
31:16	MAIN_ERR_IEN	RW	0x0	MAIN ERROR interrupt enable bit: 1: Enable interrupt 0: Disable interrupt MAIN_ERR_IEN[0]: If this bit is 1, interrupt is enabled for the 0−2KB area. … MAIN_ERR_IEN[0]: If this bit is 1, interrupt is enabled for the 30−32KB area.
15:9	RSV	-	-	Reserved
8	NVR2_ERR_IEN	RW	0x0	NVR2 ERROR interrupt enable bit: 1: Enable interrupt 0: Disable interrupt

Bit	Name	Attribute	Default Value	Functional Description
7	NVR1_ERR_IEN	RW	0x0	NVR1 ERROR interrupt enable bit: 1: Enable interrupt 0: Disable interrupt
6	NVR0_ERR_IEN	RW	0x0	NVR0 ERROR interrupt enable bit: 1: Enable interrupt 0: Disable interrupt
5	EEP2_ERR_IEN	RW	0x0	EEP2 ERROR interrupt enable bit: 1: Enable interrupt 0: Disable interrupt
4	EEP1_ERR_IEN	RW	0x0	EEP1 ERROR interrupt enable bit: 1: Enable interrupt 0: Disable interrupt
3	EEP0_ERR_IEN	RW	0x0	EEP0 ERROR interrupt enable bit: 1: Enable interrupt 0: Disable interrupt
2	KEY_ERR_IEN	RW	0x0	KEY ERROR interrupt enable bit: 1: Enable interrupt 0: Disable interrupt
1	PR_DONE_IEN	RW	0x0	Write done interrupt enable bit: 1: Enable interrupt 0: Disable interrupt
0	ER_DONE_IEN	RW	0x0	Erase done interrupt enable: 1: Enable interrupt 0: Disable interrupt

6.4.6 EFC Erase/Write Control Register (EFC_EP_CTRL) (Offset: 14H)

Bit	Name	Attribute	Default Value	Functional Description
31:10	RSV	-	-	Reserved
9	ER_T	RW	0x0	Flash erase type configuration: 0: Sector erase 1: Chip erase
8:2	RSV	-	-	Reserved

Bit	Name	Attribute	Default Value	Functional Description
1	PR_B	RW	0x0	Program request: Set by software, automatically cleared by hardware after programming is completed.
0	ER_B	RW	0x0	Erase request: Set by software, automatically cleared by hardware after erasing is completed.

6.4.7 EFC LDO TRIM Register (EFC_LDOTRIM) (Offset: 1CH)

Bit	Name	Attribute	Default Value	Functional Description
31:5	RSV	-	-	Reserved
4:0	LDO_TRIM	RW	0x10	LDO TRIM value

6.4.8 EFC RCH TRIM Register (EFC_RCHTRIM) (Offset: 20H)

Bit	Name	Attribute	Default Value	Functional Description
31:10	RSV	-	-	Reserved
9:0	RCH_TRIM	RW	0x200	60MHz RCH trim value

6.4.9 EFC RCL TRIM Register (EFC_RCLTRIM) (Offset: 24H)

Bit	Name	Attribute	Default Value	Functional Description
31:10	RSV	-	-	Reserved
9:0	RCL_TRIM	RW	0x200	32kHz RCL trim value

6.4.10 EFC PGA0 TRIM Register (EFC_PGA0TRIM) (Offset: 28H)

Bit	Name	Attribute	Default Value	Functional Description
31:13	RSV	-	-	Reserved
12:8	PGA1_TRIM	RW	0x0	-
7:5	-	-	-	Reserved
4:0	PGA0_TRIM	RW	0x0	-

6.4.11 EFC PGA1 TRIM Register (EFC_PGA1TRIM) (Offset: 2CH)

Bit	Name	Attribute	Default Value	Functional Description
31:5	RSV	-	-	Reserved
4:0	PGA2_TRIM	RW	0x0	-

6.4.12 EFC ADC TRIM Register (EFC_ADCTRIM) (Offset: 30H)

Bit	Name	Attribute	Default Value	Functional Description
31:13	RSV	-	-	Reserved
12:0	ADC_TRIM	RW	0x0	ADC TRIM value: [12]: Compensation direction 0: Positive compensation 1: Negative compensation [11:0]: ADC input compensation

6.5 Software Process

6.5.1 Write Operation

eFlash can perform write operation as it is stablized after power-on. Before performing the write operation, the corresponding key must be written to the EFC_KEY register, otherwise, the EFC will ignore this write operation.

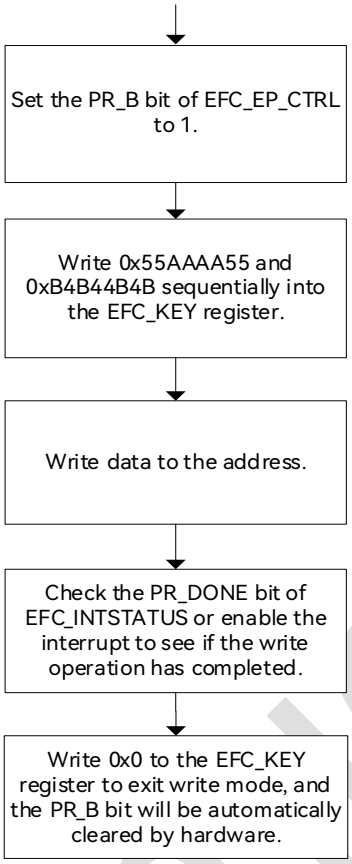


Figure 6-3: Write Operation Process

6.5.2 Erase Operation

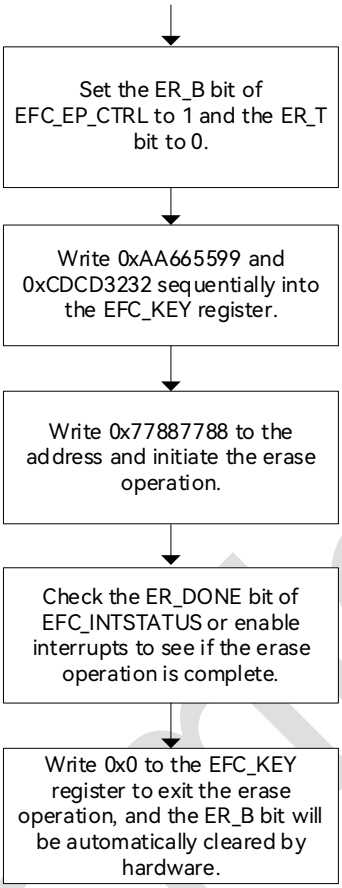


Figure 6-4: Sector Erase Operation Process

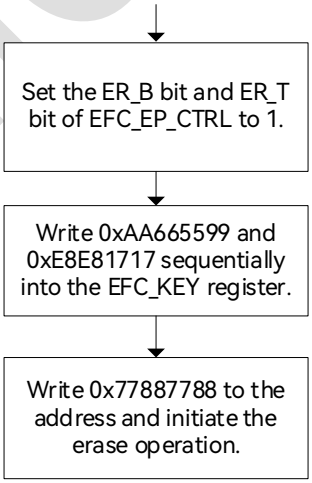


Figure 6-5: Chip Erase Operation Process

7 NVIC

7.1 Overview

The nested vectored interrupt (NVIC) is a key component of Cortex-M0+. It is tightly coupled with the CPU processor core to achieve low interrupt latency and effective handling of new incoming interrupts, which are prioritized by NVIC connecting the external interrupt signals.

Cortex-M0+ embeds an NVIC which is able to handle up to 21 interrupt request (IRQ) inputs with 4 priority levels, to handle complex logic, as well as to perform real-time control and realize interrupt processing.

All NVIC registers can only be accessed using word transfers. Any attempt to read/write half-words or bytes will result in unpredictable behavior.

The NVIC hardware block provides flexible interrupt management with minimal interrupt latency.

(For more information on NVIC, please refer to the official documentation related to Cortex-M0+ products.)

7.2 Main Features

- 32 external interrupts, each with 4 priority levels
- Interrupt wake-up controller supporting extremely-low-power Sleep mode

7.3 Interrupt Source

Table 7-1: Interrupt Source

Interrupt No.	Interrupt Source	Remarks
[0]	GPIO_PA	-
[1]	GPIO_PB	-
[2]	GPIO_PC	-
[3]	GPIO_PD	-
[4]	DMA	-
[5]	UART0	-
[6]	COMP0	-
[7]	COMP1	-
[8]	I2C	-
[9]	SPI	-
[10]	GTIMER0	-
[11]	GTIMER1	-
[12]	GTIMER2	-
[13]	LPTIMER	-
[14]	ANALOG	-
[15]	ATIMER	-
[16]	WDT	-
[17]	ADC	-
[18]	WWDT	-
[19]	UART1	-
[20]	EFC	-

8 GPIO

8.1 Overview

GPIO contains general data input and output interfaces, which can be shared with other functional pins, depending on the chip configuration. With these data interfaces, any number of pins can be configured as interrupt signals. The chip is provided with four groups of GPIOs, namely GPIOA, GPIOB, GPIOC and GPIOD, which can be abbreviated as PA, PB, PC and PD respectively. The GPIO registers shall set corresponding bits for their functions. For example, to set the direction of PA1 as output, the control bit[1] of GPIO_DIR shall be set to 1, and the settings of other bits follow this principle, that is, the direction of PAx can be set via the corresponding control bit[x] of GPIO_DIR.

8.2 Main Features

- The direction of any I/O port can be configured by software.
- Each GPIO_IN pin can be configured to trigger an interrupt in edge or level mode.
- Each GPIO_IN pin can be configured to enable or disable level filtering.

8.3 Register Description

GPIOA register base address: 0x4000_4000

GPIOB register base address: 0x4000_4400

GPIOC register base address: 0x4000_4800

GPIOD register base address: 0x4000_4C00

Table 8-1: List of GPIO Registers

Offset	Name	Description
0x00	GPIO_DIR	GPIO data direction register
0x08	GPIO_SET	GPIO output set register
0x0C	GPIO_CLR	GPIO output clear register
0x10	GPIO_ODATA	GPIO output pin mapping register
0x14	GPIO_IDATA	GPIO input pin mapping register
0x18	GPIO_IEN	GPIO interrupt enable register
0x1C	GPIO_IS	GPIO interrupt trigger mode register
0x20	GPIO_IBE	GPIO interrupt edge trigger setting register
0x24	GPIO_IEV	GPIO interrupt high/low level trigger setting register
0x28	GPIO_IC	GPIO interrupt status clear register
0x2C	GPIO_RIS	GPIO raw interrupt status register
0x30	GPIO_MIS	GPIO mask interrupt status register
0x34	GPIO_FLS	GPIO input filter selection register

8.3.1 GPIO Data Direction Register (GPIO_DIR) (Offset: 00H)

Bit	Name	Attribute	Reset Value	Description
7:0	DIR	RW	0x0	8-bit GPIO input/output control register: 0: Input 1: Output

Note: The 8 bits in the GPIOx_DIR[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_DIR[1] corresponds to PA1.

8.3.2 GPIO Output Set Register (GPIO_SET) (Offset: 08H)

Bit	Name	Attribute	Reset Value	Description
7:0	SET	W	0x0	8-bit GPIO output set register: 0: Invalid operation 1: IO is set to 1 when IO is output.

Note: The 8 bits in the GPIOx_SET[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_SET[1] corresponds to PA1.

8.3.3 GPIO Output Clear Register (GPIO_CLR) (Offset: 0CH)

Bit	Name	Attribute	Reset Value	Description
7:0	CLR	W	0x0	8-bit GPIO output clear register: 0: Invalid operation 1: IO is cleared when IO is output.

Note: The 8 bits in the GPIOx_CLR[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_CLR[1] corresponds to PA1.

8.3.4 GPIO Output Pin Mapping Register (GPIO_ODATA) (Offset: 10H)

Bit	Name	Attribute	Reset Value	Description
7:0	ODATA	RW	0x0	8-bit GPIO output pin mapping register: When the GPIO direction is output active, the write operation is directed at the external pins, and the read operation is performed to obtain the external pin value.

Note: The 8 bits in the GPIOx_ODATA[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_ODATA[1] corresponds to PA1.

8.3.5 GPIO Input Pin Mapping Register (GPIO_IDATA) (Offset: 14H)

Bit	Name	Attribute	Reset Value	Description
7:0	IDATA	R	0x0	8-bit GPIO input pin mapping register: When the GPIO direction is input active, read operations retrieve the external pin value; this register is read-only.

Note: The 8 bits in the GPIOx_IDATA[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_IDATA[1] corresponds to PA1.

8.3.6 GPIO Interrupt Enable Register (GPIO_IEN) (Offset: 18H)

Bit	Name	Attribute	Reset Value	Description
7:0	IEN	RW	0x0	8-bit GPIO interrupt enable register: 0: Disable corresponding pin interrupt 1: Enable corresponding pin interrupt

Note: The 8 bits in the GPIOx_IEN[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_IEN[1] corresponds to PA1.

8.3.7 GPIO Interrupt Trigger Mode Register (GPIO_IS) (Offset: 1CH)

Bit	Name	Attribute	Reset Value	Description
7:0	IS	RW	0x0	7-bit GPIO interrupt trigger mode register: 0: Edge trigger 1: Level trigger

Note: The 8 bits in the GPIOx_IS[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_IS[1] corresponds to PA1.

8.3.8 GPIO Interrupt Edge Trigger Setting Register (GPIO_IBE) (Offset: 20H)

Bit	Name	Attribute	Reset Value	Description
7:0	IBE	RW	0x0	8-bit GPIO interrupt edge trigger setting register: 0: Single-edge trigger 1: Dual-edge trigger

Note: The 8 bits in the GPIOx_SET[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_SET[1] corresponds to PA1.

8.3.9 GPIO Interrupt High/Low Level Trigger Setting Register (GPIO_IEV) (Offset: 24H)

Bit	Name	Attribute	Reset Value	Description
7:0	IEV	RW	0x0	8-bit GPIO interrupt high/low level trigger setting register: 0: Falling-edge/low-level trigger 1: Rising-edge/high-level trigger

Note: The 8 bits in the GPIOx_IEV[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_IEV[1] corresponds to PA1.

8.3.10 GPIO Interrupt Status Clear Register (GPIO_IC) (Offset: 28H)

Bit	Name	Attribute	Reset Value	Description
7:0	IC	W	0x0	8-bit GPIO interrupt clear register: 0: Invalid operation 1: Clear interrupt of the corresponding pin

Note: The 8 bits in the GPIOx_IC[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_IC[1] corresponds to PA1.

8.3.11 GPIO Raw Interrupt Status Register (GPIO_RIS) (Offset: 2CH)

Bit	Name	Attribute	Reset Value	Description
7:0	RIS	R	0x0	8-bit GPIO raw interrupt register: 0: No interrupt pending on the corresponding pin 1: With interrupt pending on the corresponding pin

Note: The 8 bits in the GPIOx_RIS[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_RIS[1] corresponds to PA1.

8.3.12 GPIO Mask Interrupt Status Register (GPIO_MIS) (Offset: 30H)

Bit	Name	Attribute	Reset Value	Description
7:0	MIS	R	0x0	32-bit GPIO mask interrupt status register: reflecting the masked interrupt status of the corresponding pin.

Note: The 8 bits in the GPIOx_MIS[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_MIS[1] corresponds to PA1.

8.3.13 GPIO Input Filter Selection Register (GPIO_FLS) (Offset: 34H)

Bit	Name	Attribute	Reset Value	Description
7:0	FLS	RW	0x0	8-bit GPIO input filter selection register: indicating whether filtering is required for the corresponding pin input. 1: Enable filtering 0: Disable filtering

Note: The 8 bits in the GPIOx_FLS[y] (x = A…D) register correspond to the 8 pins in this group respectively. For example, GPIOA_FLS[1] corresponds to PA1.

8.4 Operation Procedure

8.4.1 IO

1. Configure the GPIO_DIR register to select the GPIO direction.
2. Use GPIO_SET/GPIO_CLR or GPIO_ODATA to set the output level.
3. Configure GPIO_FLS to determine whether filtering is required for the input level.
4. Use GPIO_IDATA to obtain the input pin level.

8.4.2 Interrupt Trigger Mode

Interrupt initialization process:

1. Set GPIO_DIR as input.
2. Clear GPIO_IEN to avoid exceptions.
3. Configure the GPIO_IS register to select the edge or level trigger mode.
4. In single-edge trigger mode, configure the GPIO_IBE register to determine whether it is single-edge trigger or dual-edge trigger.
5. In single-edge trigger mode, configure the GPIO_IEV register to determine whether it is rising or falling edge trigger.
6. In level trigger mode, configure the GPIO_IEV register to determine whether it is high or low level trigger.
7. Configure the GPIO_IC register to clear the interrupt.
8. Configure the GPIO_IEN register to enable the interrupt of corresponding bit.

8.4.3 Clearing Interrupt

ISR writes GPIO_IC to clear the interrupt status. If a new edge-triggered interrupt is generated while clearing the register, this new interrupt will remain valid until the next clear. The interrupt status shall be read before clearing GPIO_IEN, which will clear the corresponding interrupt status.

8.4.4 Input Filtering

In non-low power mode, the GPIO_FLS register can be configured to determine if digital filtering is required for the corresponding pin input level. If disabled, the level signal will be output directly; if enabled, the input level must be stable for at least 3 system clock cycles before it is detected inside the GPIO.

9 UART0/1

9.1 Overview

UART0 and UART1 are serial port modules that support full-duplex data transfer and serial communication with external interface devices.

9.2 Main Features

- Providing standard asynchronous communication bits (start bit, parity bit, stop bit)
  - 1 start bit
  - Data bit width of 8 bits
  - 1 parity bit (odd or even), or no parity bit
  - 1 or 2 stop bits
  - Bytes transmitted sequentially from LSB to MSB
- Programmable baud rate
- Supporting transfer at common baud rates such as 9600 bps, 19200 bps and 115200 bps
- UART0 supports single-wire transfer on TX port

9.3 Register Description

UART0 register base address: 0x4000_0000

UART1 register base address: 0x4000_0040

Table 9-1: List of UART0/1 Registers

Offset	Name	Description
0x00	UART_ISR	Interrupt status register

Offset	Name	Description
0x04	UART_IER	Interrupt enable register
0x08	UART_CR	Control register
0x0C	UART_TDR	Transmit data register
0x10	UART_RDR	Receive data register
0x14	UART_BRR	Baud rate parameter low register
0x18	UART_DLTOR	Delay timeout register

9.3.1 Interrupt Status Register UART_ISR (Offset: 00H)

Bit	Name	Attribute	Reset Value	Description
31:19	RSV	-	-	Reserved
18	PE	W1C	0x0	Parity error flag, cleared by software via writing it to 1
17	FE	W1C	0x0	Frame format error flag, cleared by software via writing it to 1
16	OVERE	W1C	0x0	Receive buffer overflow error flag, cleared by software via writing it to 1 or by reading RDR
15:12	RSV	-	-	Reserved
11	RXTO	W1C	0x0	Receive timeout flag, cleared by software via writing it to 1
10:9	RSV	-	-	Reserved
8	RXF	W1C	0x0	Receive buffer full flag, cleared by software via writing it to 1 or by reading RDR
7	NEGWKF	W1C	0x0	Falling edge wakeup flag, cleared by software via writing it to 1
6:2	RSV	-	-	Reserved
1	TXE	R	0x1	Transmit buffer empty flag, cleared by software via writing data to TDR
0	TC	W1C	0x0	Transmission complete flag, cleared by software via writing it to 1 or writing data to TDR

9.3.2 Interrupt Enable Register UART_IER (Offset: 04H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
11	RXTOIE	RW	0x0	Receive timeout interrupt enable: 1: Enabled
10	RXERRIE	RW	0x0	Receive error interrupt enable: 1: Enabled
9	RSV	-	-	Reserved
8	RXFIE	RW	0x0	Receive buffer full interrupt enable: 1: Enabled
7	NEGWKIE	RW	0x0	Falling edge wakeup enable: 1: Enabled
6:2	RSV	-	-	Reserved
1	TXIE	RW	0x0	Transmit buffer empty interrupt enable: 1: Enabled
0	TCIE	RW	0x0	Transmission complete interrupt enable: 1: Enabled

9.3.3 Control Register UART_CR (Offset: 08H)

Bit	Name	Attribute	Reset Value	Description
31:14	RSV	-	-	Reserved
13	BUSY	R	0x0	UART data transmission busy flag: 0: Idle 1: UART is transmitting data.
12	RXTOEN	RW	0x0	Receive data timeout enable: 0: Disabled 1: Enabled
11	NEGEWKE	RW	0x0	Falling edge wakeup enable: 0: Disabled 1: Enabled
10	TX_EN	RW	0x0	UART single-wire mode enable: 0: Disabled 1: Enabled
9	TX_OEN	RW	0x0	UART TX pin data transmission direction control in single-wire mode: 0: TX pin as data output 1: TX pin as data input

Bit	Name	Attribute	Reset Value	Description
8	STOPLEN	RW	0x0	Stop bit width: 0: 1 stop bit 1: 2 stop bits
7:6	DATALEN	RW	0x1	Data length per frame: 00: 7 data bits 01: 8 data bits 10: 9 data bits 11: Reserved
5:4	PARITYSL	RW	0x0	Parity bit configuration: 00: No parity bit 01: Even parity 10: Odd parity 11: Reserved
3	RXPOL	RW	0x0	Receive data polarity configuration: 0: Normal 1: Inverted
2	TXPOL	RW	0x0	Transmit data polarity configuration: 0: Normal 1: Inverted
1	RXEN	RW	0x0	Receive enable: 1: Enabled
0	TXEN	RW	0x0	Transmit enable: 1: Enabled

9.3.4 Transmit Data Register UART_TDR (Offset: 0CH)

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8:0	TDR	W	0x0	Store the data to be transmitted.

9.3.5 Receive Data Register UART_RDR (Offset: 10H)

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8:0	RDR	R	0x0	Store the data to be received.

9.3.6 Baud Rate Parameter Low Register UART_BRR (Offset: 14H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	BRR	RW	0x340	Baud rate generator register value Baud rate calculation formula: When BRR < 16, F = UART_CLK / 16 When BRR ≥ 16, F = UART_CLK / (BRR + 1)

9.3.7 Delay Timeout Register UART_DLTOR (Offset: 18H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	TXDLY	RW	0x0	Data transmission delay time, measured in clock cycles
7:0	RXDLY	RW	0xFF	Data reception timeout, measured in clock cycles

9.4 Operation Procedure

9.4.1 Transmission and Reception via UART

1. Configure the system configuration register for the UART module clock.
2. Configure the system configuration register for the UART alternate function.
3. Configure UART interrupts.
4. Configure the UART interrupt enable register (whether interrupt is adopted).
5. Configure the UART control register (parity bit, etc.).
6. Configure the baud rate.
7. Enable the UART.

9.4.2 UART Initialization

1. Clear the UART_ISR register by writing 0.
2. Configure the interrupt enable register UART_IER to enable or disable the generation of corresponding interrupt pulses.
3. Clear the UART_CR register by writing 0.
4. Configure the UART_BRR register.

9.4.3 UART Transmitting Byte

1. Before transmitting and receiving data, software can configure the baud rate parameters, parity type and interrupt enable.
2. Set UART_CR.TXEN = 1.
3. Write the first byte of data to UART_TDR.
4. Query the transmit completion flag UART_ISR.TC, if TC = 1, the current data has been transmitted. Clear this bit by writing 1.
5. Continue writing the next byte to UART_TDR.

9.4.4 UART Receiving Byte

1. Before transmitting and receiving data, software can configure the baud rate parameter, parity type and interrupt enable.
2. To receive data, check the UART_ISR flag bits or wait for an interrupt.
3. Read the UART_RDR register.
4. Continue receiving data.

10 I2C

10.1 Overview

The I2C bus interface handles communications between the microcontroller and the serial I2C bus. The I2C module is responsible for receiving and transmitting data, as well as converting data between serial and parallel formats. It is connected to the I2C bus via the data line SDA and the clock line SCL, controlling all the I2C bus-specific sequencing. This module supports both master and slave modes.

10.2 Main Features

- Standard-mode (up to 100 kbps) / fast-mode (up to 400 kbps) / fast-mode plus (up to 1 Mbps)
- 7-bit and 10-bit addressing modes
- Interrupt polling function
- Low-power slave design, capable of receiving and transmitting data without a system clock
- Supporting asynchronous slave address matching wake-up, data frame reception completion wake-up, or START detection wake-up.

10.3 Register Description

I2C register base address: 0x4000_5400

Table 10-1: List of I2C Registers

Offset	Name	Description
0x00	I2C_MCFG	I2C master configuration register
0x04	I2C_MCON	I2C master control register
0x08	I2C_MIE	I2C master interrupt enable register

Offset	Name	Description
0x0C	I2C_MIF	I2C master interrupt flag register
0x10	I2C_MSTA	I2C master status register
0x14	I2C_MBRG	I2C master baud rate register
0x18	I2C_MBUF	I2C master transmit/receive buffer register
0x1C	I2C_MTIM	I2C master timing control register
0x20	I2C_MTO	I2C master timeout register
0x24	I2C_SCON	I2C slave control register
0x28	I2C_SIE	I2C slave interrupt enable register
0x2C	I2C_SIF	I2C slave interrupt flag register
0x30	I2C_SSTA	I2C slave status register
0x34	I2C_SBUF	I2C slave transmit/receive buffer register
0x38	I2C_SADR0	I2C slave address register 0
0x3C	I2C_SADR1	I2C slave address register 1
0x40	I2C_SADR2	I2C slave address register 2
0x44	I2C_SADR3	I2C slave address register 3

10.3.1 I2C Master Configuration Register I2C_MCFG (Offset: 00H)

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3	MAUST	RW	0x0	Configuration bit for automatic STOP after DMA transfer: 1: After the DMA transfer of the specified length is completed, the STOP sequence is automatically sent. 0: After the DMA transfer of the specified length is completed, wait for the software to take over.
2	MDMAEN	RW	0x0	Master DMA enable: 0: DMA function disabled 1: DMA function enabled
1	MTOEN	RW	0x0	SCL low timeout enable: 1: Timeout function enabled 1: Timeout function disabled
0	MSPEN	RW	0x0	I2C master module enable bit: 1: Master module enabled 0: Master module disabled

10.3.2 I2C Master Control Register I2C_MCON (Offset: 04H)

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3	MREN	RW	0x0	Receive enable in master receive mode: 1: Master receive enabled 0: Master receive disabled During master communication, after the software has sent the address byte, it sets the MREN bit to switch the transfer direction to master receive, allowing it to receive data from the slave.
2	MSTS	RW	0x0	STOP timing generation enable bit: the software writes 1 to send the STOP timing, and the hardware automatically clears it after the transmission is completed.
1	MRSTRTS	RW	0x0	Repeat START timing generation enable bit: the software writes 1 to send the Repeat START timing, and the hardware automatically clears it after the transmission is completed.
0	MSTRTS	RW	0x0	START timing generation enable bit: the software writes 1 to send the START timing, and the hardware automatically clears it after the transmission is completed.

10.3.3 I2C Master Interrupt Enable Register I2C_MIE (Offset: 08H)

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	MWCIE	RW	0x0	MWCIF interrupt enable register: 1: Enable write conflict interrupt 0: Disable write conflict interrupt
5	MTOIE	RW	0x0	SCL timeout interrupt enable register: 1: Enable timeout interrupt 0: Disable timeout interrupt
4	MSIE	RW	0x0	START timing interrupt enable register: 1: Enable START timing interrupt

Bit	Name	Attribute	Reset Value	Description
				0: Disable START timing interrupt
3	MSPIE	RW	0x0	STOP timing interrupt enable register: 1: Enable STOP timing interrupt 0: Disable STOP timing interrupt
2	MNAIE	RW	0x0	Master transmit mode NACK interrupt enable register: 1: Enable interrupt on receiving NACK 0: Disable interrupt on receiving NACK
1	MTXIE	RW	0x0	I2C master transmit complete interrupt enable: 1: Enable transmit complete interrupt 0: Disable transmit complete interrupt
0	MRXIE	RW	0x0	I2C master receive complete interrupt enable: 1: Enable receive complete interrupt 0: Disable receive complete interrupt

10.3.4 I2C Master Interrupt Flag Register I2C_MIF (Offset: 0CH)

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	MWCIF	W1C	0x0	Write conflict detection bit: The MCU can only write to MBUF after completing the START timing or after transmitting a frame of read/write data; otherwise, a write conflict occurs. This bit is set by hardware and cleared by software writing 1. 1: Write conflict occurred 0: No conflict occurred
5	MTOIF	W1C	0x0	SCL timeout interrupt flag, which only works when TOEN is 1, and is cleared by software writing 1. 1: SCL timeout occurred 0: No SCL timeout occurred
4	MSIF	W1C	0x0	START timing transmission complete interrupt flag, set by hardware and cleared by software writing 1
3	MSPIF	W1C	0x0	STOP timing transmission complete interrupt flag, set by hardware and cleared by software writing 1
2	MNAIF	W1C	0x0	Response signal from slave in master transmit

Bit	Name	Attribute	Reset Value	Description
				mode: when the master receives an NACK after transmission, this flag can generate an interrupt. It is set by hardware and cleared by software writing 1. 1: Slave responds with NACK 0: Slave responds with ACK
1	MTXIF	W1C	0x0	I2C master transmit complete interrupt flag, set by hardware and cleared by software writing 1. This flag register is set after the master receives an ACK or NACK from the slave.
0	MRXIF	W1C	0x0	I2C master receive complete interrupt flag, set by hardware and cleared by software writing 1. This flag register is set after the master responds with an ACK or NACK.

10.3.5 I2C Master Status Register I2C_MSTA (Offset: 10H)

Bit	Name	Attribute	Reset Value	Description
31:6	RSV	-	-	Reserved
5	MBUSY	R	0x0	I2C communication status bit: 1: The interface is in read/write state, with data transfer in progress. 0: Data transfer has been completed.
4	MRW	R	0x0	I2C transfer direction status bit: 1: The master is reading data from the slave. 0: The master is writing data to the slave.
3	RSV	-	-	Reserved
2	MBUFF	R	0x0	Buffer full status bit: Receiving: 1: Reception complete, MBUF is full. 0: Reception incomplete, MBUF is empty. Transmitting: 1: Transmitting, MBUF is full. 0: Transmission complete, MBUF is empty.
1	RSV	-	-	Reserved
0	MACKST	RW	0x0	Status of master response signal in master

Bit	Name	Attribute	Reset Value	Description
				receive mode: 1: The master responds with NACK. 0: The master responds with ACK.

10.3.6 I2C Master Baud Rate Register I2C_MBRG (Offset: 14H)

Bit	Name	Attribute	Reset Value	Description
31:25	RSV	-	-	Reserved
24:16	MBRGH	RW	0x13	SCL clock high level width sent by the master, counted in PCLK cycles
15:9	RSV	-	-	Reserved
8:0	MBRGL	RW	0x13	SCL clock low level width sent by the master, counted in PCLK cycles

10.3.7 I2C Master Transmit and Receive Buffer Register I2C_MBUF (Offset: 18H)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	MBUF	RW	0x0	I2C master data transfer register

10.3.8 I2C Master Timing Control Register I2C_MTIM (Offset: 1CH)

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
8:0	MDY	RC	0x4	Define the hold time parameter of SDA relative to the falling edge of SCL, counted in I2C working clock cycles. Note: The minimum valid value is 1, and the maximum valid value is MBRGL.

10.3.9 I2C Master Timeout Register I2C_MTO (Offset: 20H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	MTO	RW	0xFFF	Define the timeout period for SCL low level extension in the I2C slave, which can be rewritten by software when MSPEN is set to 0.

10.3.10 I2C Slave Control Register I2C_SCON (Offset: 24H)

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	SSCLLEN	RW	0x1	I2C slave clock stretching enable: 1: Disable slave clock stretching 1: Enable slave clock stretching
5	SDMAEN	RW	0x0	I2C slave DMA enable: 1: Enable DMA function 0: Disable DMA function
4	SACKEN	RW	0x1	ACK enable: 1: Slave will respond with ACK after receiving data. 0: Slave will not respond with ACK.
3	SSDADYEN	RW	0x0	SDA slave output delay enable: 1: Enable slave SDA output delay 0: Disable slave SDA output delay
2	SSCLLVEN	RW	0x0	SCL slave input analog filtering enable: 1: Enable analog filtering 0: Bypass analog filtering
1	AD10EN	RW	0x0	10-bit addressing enable: 1: Slave adopts 10-bit addressing 0: Slave adopts 7-bit addressing
0	SSPEN	RW	0x0	I2C slave enable: 1: Enable I2C slave 0: Disable I2C slave

10.3.11 I2C Slave Interrupt Enable Register I2C_SIE (Offset: 28H)

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12	AD3IE	RW	0x0	Slave address 3 match interrupt enable: 1: Enable interrupt on address 3 match 0: Disable interrupt on address 3 match
11	AD2IE	RW	0x0	Slave address 2 match interrupt enable: 1: Enable interrupt on address 2 match 0: Disable interrupt on address 2 match
10	AD1IE	RW	0x0	Slave address 1 match interrupt enable: 1: Enable interrupt on address 1 match 0: Disable interrupt on address 1 match
9	AD0IE	RW	0x0	Slave address 0 match interrupt enable: 1: Enable interrupt on address 0 match 0: Disable interrupt on address 0 match
8	RSV	-	-	Reserved
7	ADER	RW	0x0	Slave address error interrupt enable: 1: Enable address error interrupt 0: Disable address error interrupt
6	SSIE	RW	0x0	Slave start interrupt enable: 1: Enable start interrupt 0: Disable start interrupt
5	SSPIE	RW	0x0	Slave stop interrupt enable: 1: Enable stop interrupt 0: Disable stop interrupt
4	SWCIE	RW	0x0	Slave SWCIF interrupt enable: 1: Enable write conflict interrupt 0: Disable write conflict interrupt
3	SSOVIE	RW	0x0	Slave receive overflow interrupt enable: 1: Enable receive overflow interrupt 0: Disable receive overflow interrupt
2	SADIE	RW	0x0	I2C slave address match interrupt enable: 1: Enable interrupt on address match 0: Disable interrupt on address match
1	STXIE	RW	0x0	I2C slave transmit complete interrupt enable: 1: Enable transmit complete interrupt

Bit	Name	Attribute	Reset Value	Description
				0: Disable transmit complete interrupt
0	SRXIE	RW	0x0	I2C slave receive complete interrupt enable: 1: Enable receive complete interrupt 0: Disable receive complete interrupt

10.3.12 I2C Slave Interrupt Status Register I2C_SIF (Offset: 2CH)

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12	AD3IF	W1C	0x0	Slave address 3 match flag: 1: The received 7-bit address matches the content of the I2C_SADR3 register. 0: The received address does not match the I2C_SADR3 register. This flag bit is set by hardware and cleared by software writing 1.
11	AD2IF	W1C	0x0	Slave address 2 match flag: 1: The received 7-bit address matches the content of the I2C_SADR2 register. 0: The received address does not match the I2C_SADR2 register. This flag bit is set by hardware and cleared by software writing 1.
10	AD1IF	W1C	0x0	Slave address 1 match flag: 1: The received 7-bit address matches the content of the I2C_SADR1 register. 0: The received address does not match the I2C_SADR1 register. This flag bit is set by hardware and cleared by software writing 1.
9	AD0IF	W1C	0x0	Slave address 0 match flag: 1: The received 7-bit or 10-bit address matches the content of the I2C_SADR0 register. 0: The received address does not match the I2C_SADR0 register. This flag bit is set by hardware and cleared by software writing 1.

Bit	Name	Attribute	Reset Value	Description
8	RSV	-	-	Reserved
7	ADER	W1C	0x0	Slave address format error: 1: An address format error is triggered when a received address byte starts with 11110 in 7-bit address mode, or the first byte does not start with 11110 in 10-bit address mode. 0: No frame format error detected This flag bit is set by hardware and cleared by software writing 1.
6	SSIF	RC	0x0	START interrupt bit: 1: Slave detects a START condition. 0: Slave does not detect a START condition. It is set by hardware and cleared by software read.
5	SSPIF	RC	0x0	STOP interrupt bit: 1: Slave detects a STOP condition. 0: Slave does not detect a STOP condition. It is set by hardware and cleared by software read.
4	SWCIF	W1C	0x0	Write conflict flag: 1: Software writes new data to SSPBUF while SBUFF = 1. 0: No write conflict When an overflow occurs, the new data will be discarded. This flag bit is set by hardware and cleared by software writing 1.
3	SSOVIF	W1C	0x0	SBUF overflow flag: 1: Slave receives new data while SBUFF = 1. 0: No receive overflow If slave SCL stretching is enabled, receive data overflow will not occur. This flag bit is set by hardware and cleared by software writing 1.
2	SADIF	W1C	0x0	Slave address match flag: 1: The received 7-bit or 10-bit address matches the content of the I2C_SADRx register.

Bit	Name	Attribute	Reset Value	Description
				0: The received address does not match the I2C_SADRx register. This flag bit is set by hardware and cleared by software writing 1.
1	STXIF	W1C	0x0	Slave transmission complete flag: 1: Transmission is completed. 0: Transmission is not completed. This flag bit is set by hardware and cleared by software writing 1.
0	SRXIF	W1C	0x0	Slave reception complete flag: 1: Reception is completed. 0: Reception is not completed. This flag bit is set by hardware and cleared by software writing 1.

10.3.13 I2C Slave Status Register I2C_SSTA (Offset: 30H)

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3	SBUSY	R	0x0	Slave communication flag: 1: Slave is in data transmission or reception. 0: Slave is idle.
2	SRW	R	0x0	Read/write direction status register: 1: Slave receives RW = 1, indicating the slave needs to send data to the master. 0: Slave is in data reception state.
1	DA	R	0x0	Frame indication: 1: The last byte received was data. 0: The last byte received was an address.
0	SBUFF	R	0x0	Slave data buffer full flag: 1: SBUFF is full. 0: SBUFF is empty.

10.3.14 I2C Slave Transmit and Receive Buffer Register I2C_SBUF (Offset: 34H)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	SBUF	RW	0x0	I2C slave data transfer register

10.3.15 I2C Slave Address Register 0 I2C_SADR0 (Offset: 38H)

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9:0	SADR0	RW	0x0	I2C slave address register 0

10.3.16 I2C Slave Address Register 1 I2C_SADR1 (Offset: 3CH)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:1	SADR1	RW	0x0	I2C slave address register 1
0	SADR1EN	RW	0x0	1: SADR1 enabled; 0: SADR1 not enabled

10.3.17 I2C Slave Address Register 2 I2C_SADR2 (Offset: 40H)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:1	SADR2	RW	0x0	I2C slave address register 2
0	SADR2EN	RW	0x0	1: SADR2 enabled; 0: SADR2 not enabled

10.3.18 I2C Slave Address Register 3 I2C_SADR3 (Offset: 44H)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:1	SADR3	RW	0x0	I2C slave address register 3
0	SADR3EN	RW	0x0	1: SADR3 enabled; 0: SADR3 not enabled

10.4 Operation Procedure

10.4.1 Master Transmit

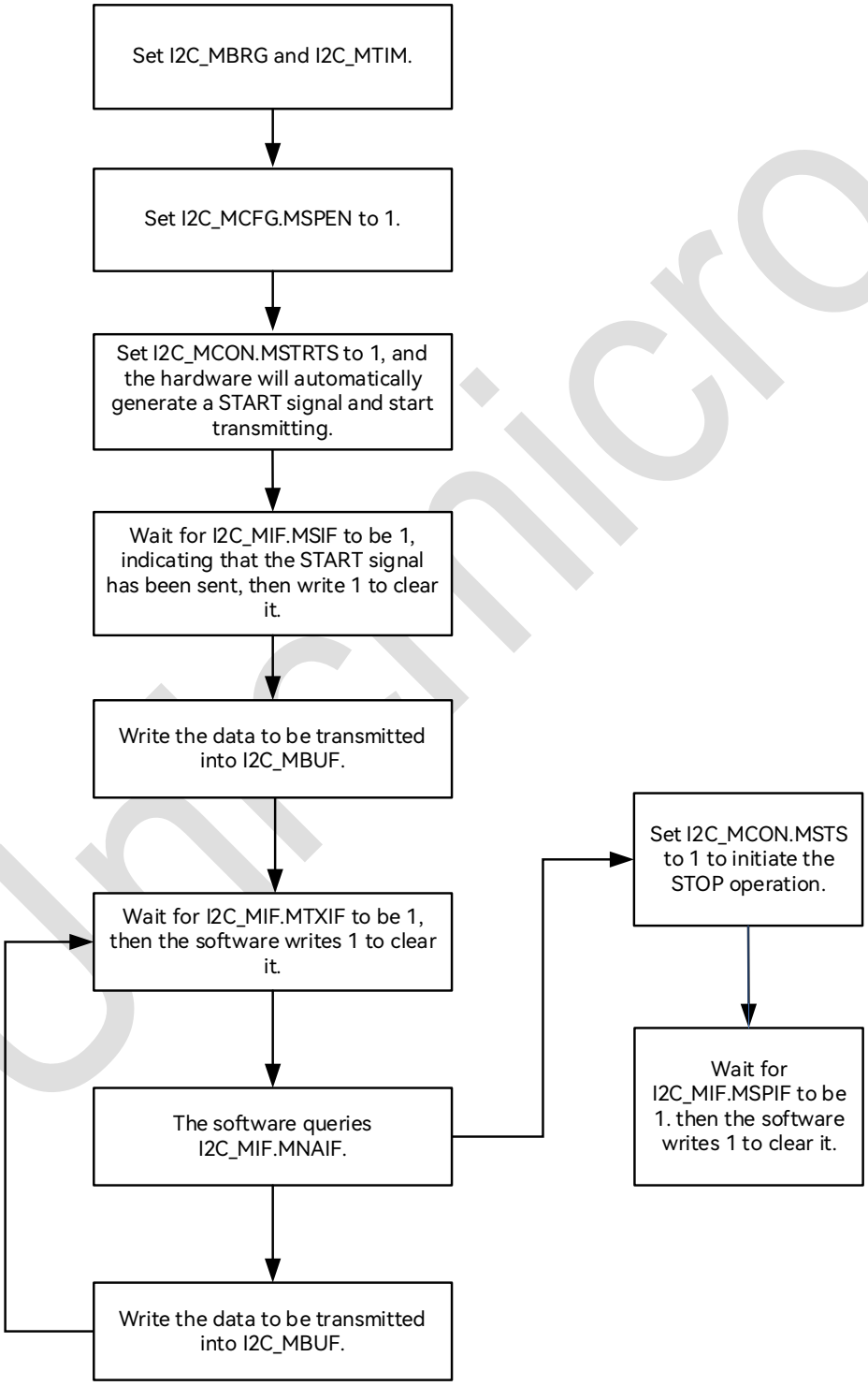


Figure 10-1: Master Transmit Flowchart

10.4.2 Master Receive

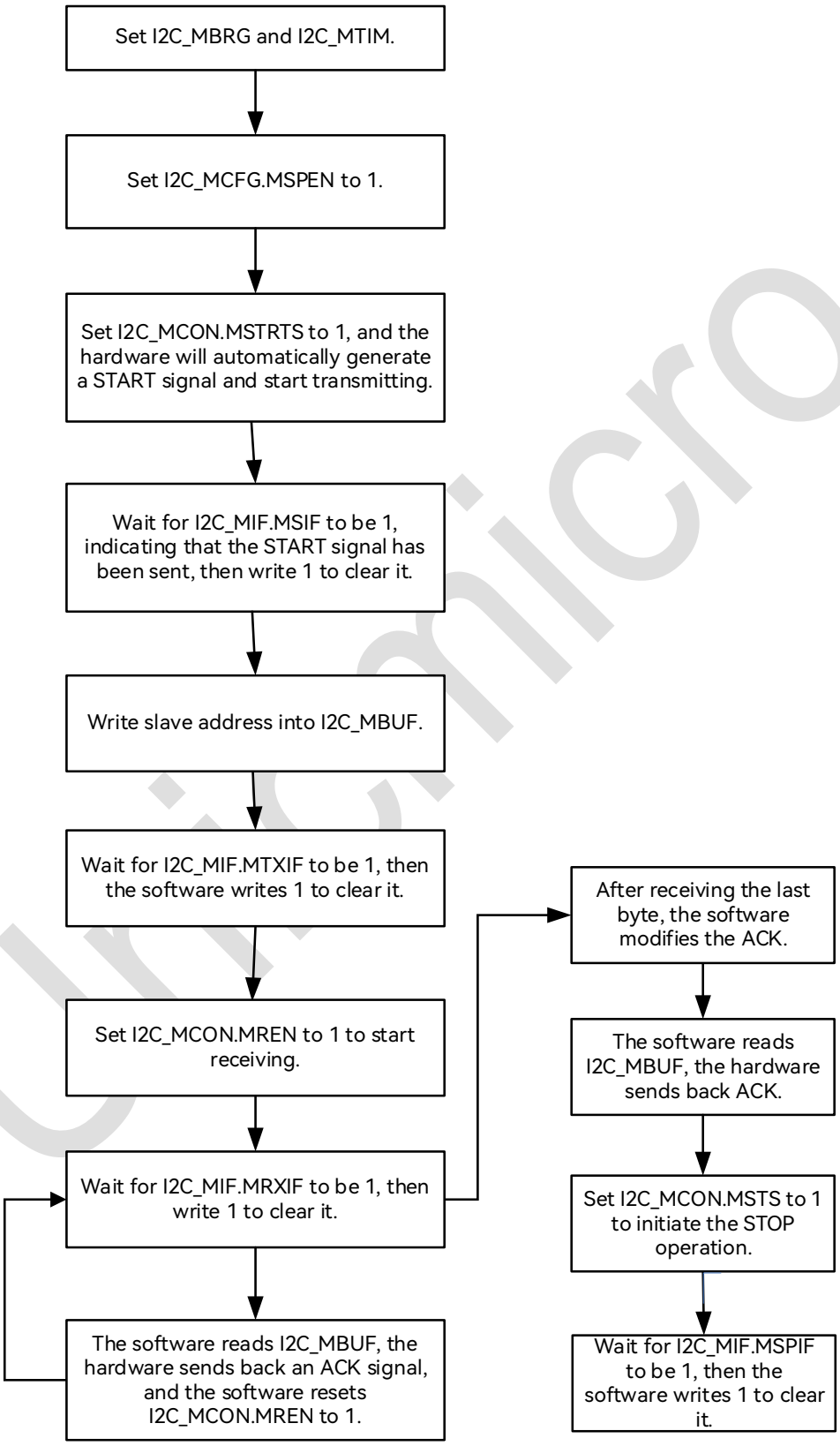


Figure 10-2: Master Receive Flowchart

10.4.3 Slave Transmit

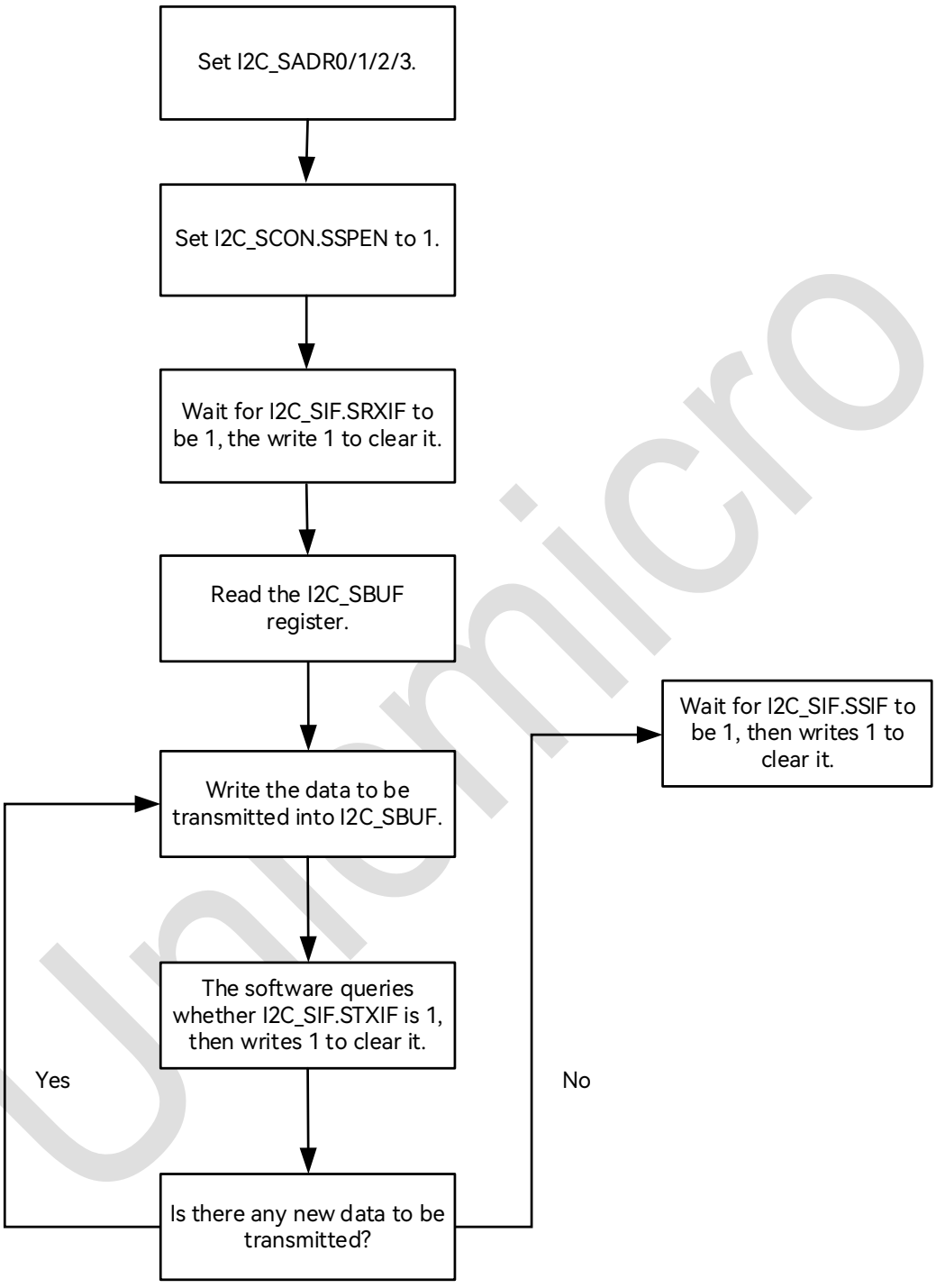


Figure 10-3: Slave Transmit Flowchart

10.4.4 Slave Receive

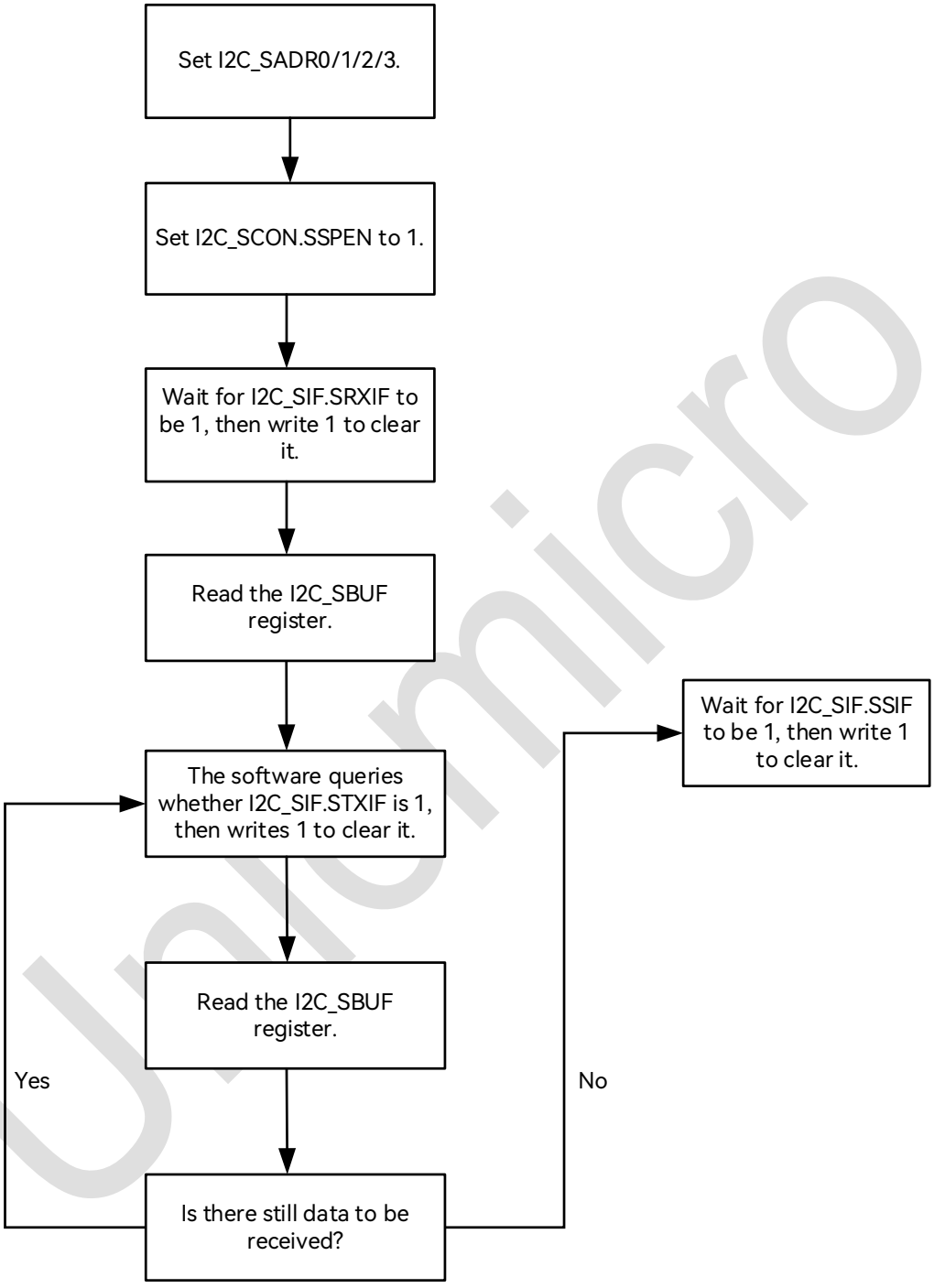


Figure 10-4: Slave Receive Flowchart

11 SPI

11.1 Overview

The serial peripheral interface (SPI) is a serial synchronous communication means for external devices to exchange data over a single line. The chip provides one SPI module that can be configured as a master or slave device to enable SPI communication with the outside.

11.2 Main Features

- Full-duplex or half-duplex 3/4-wire serial synchronous transmission and reception
- Master and slave modes
- Programmable clock polarity and phase (supporting modes 0/1/2/3)
- Programmable bit rate
- Slave mode frequency up to $f_{sys}/2$
- Data transfer of 6/7/8/9 bits
- Supporting DC control
- Transfer complete interrupt flag
- Write conflict error flag
- Error detection, protection and interrupt flags in master mode
- Supporting DMA
- 9-bit TX and RX FIFOs with a depth of 8

11.3 Register Description

SPI register base address: 0x4000_0800

Table 11-1: List of SPI Registers

Offset	Name	Description
0x00	SPI_CR	SPI configuration register
0x04	SPI_CS0	SPI master mode control register 0
0x08	SPI_CS1	SPI master mode control register 1
0x14	SPI_OPCR	SPI process control register
0x18	SPI_IE	SPI interrupt enable register
0x1C	SPI_IF	SPI interrupt flag register
0x20	SPI_TXBUF	SPI transmit buffer register
0x24	SPI_RXBUF	SPI receive buffer register
0x28	SPI_DMARXLEV	SPI DMA receive setting register
0x2c	SPI_DMATXLEV	SPI DMA transmit setting register

11.3.1 SPI Configuration Register (SPI_CR) (Offset: 00H)

Bit	Name	Attribute	Reset Value	Description
31:20	RSV	-	-	Reserved
19	MOSI_POL	RW	0x0	In SPI master mode, default idle level for MOSI: 1: MOSI is high when idle. 0: MOSI is low when idle. Note: The value of this bit cannot be modified while communication is in progress or while SSN is low.
18	DC_REG	RW	0x0	DC output register control bit: 1: DC output is 1. 0: DC output depends on the 8th bit in FIFO.
17	DC_DIR	RW	0x0	DC direction control: 1: DC is in output mode. 0: DC is in input mode.
16	DC_EN	RW	0x0	DC control enable: 1: Enable DC mode, DC is multiplexed with MOSI. 0: Disable DC mode.

Bit	Name	Attribute	Reset Value	Description
15:14	SPI_FRS	RW	0x2	SPI frame format setting: 0: 6-bit format 1: 7-bit format 2: 8-bit format 3: 9-bit format
13	DMA_TX_EN	RW	0x0	DMA TX enable: 1: Enable DMA TX request 0: Disable DMA TX request
12	DMA_RX_EN	RW	0x0	DMA RX enable: 1: Enable DMA RX request 0: Disable DMA RX request
11	FLTEN	RW	0x1	Slave input pin (SSN/SCK/MOSI) filtering enable: 1: Enable 4ns filtering 0: No filtering
10	SSNM	RW	0x0	SSN control mode selection in master mode: 1: After transmitting 8 bits, the master asserts SSN high, with the high time controlled by the WAIT register. 0: After transmitting 8 bits, the master keeps SSN low, with the low time controlled by the WAIT register.
9	TXO_AC	RW	0x1	TXONLY hardware auto-clear enable: 1: TXONLY hardware auto-clear is enabled; after software enables TXO, hardware clears it after transmission is complete. 0: Disable TXONLY hardware auto-clear
8	TXO	RW	0x0	TXONLY control bit: 1: Enable single-transmit mode for the master 0: Disable single-transmit mode
7	MSPA	RW	0x0	Master sampling position adjustment: master adjusts the sampling position of MISO signal to compensate PCB routing delay during high-speed communication: 1: Sampling point is delayed by half an SCK cycle. 0: No adjustment

Bit	Name	Attribute	Reset Value	Description
6	SSPA	RW	0x0	Slave sending position adjustment: slave adjusts the MISO sending position: 1: Send MISO half an SCK cycle in advance. 0: No adjustment
5	MM	RW	0x0	Master/slave mode selection: 1: Master mode 0: Slave mode
4:3	WAIT	RW	0x0	In master mode, after transmitting 8 bits, an additional delay of at least (1 + WAIT) SCK cycles shall be introduced before transmitting the next 8 bits of data.
2	RSV	-	-	RFU: not implemented, read as 0
1	SSNSEN	RW	0x0	Software-controlled SSN enable in master mode: 1: In master mode, SSN output is controlled by software. 0: In master mode, SSN output is automatically controlled by hardware; SSN0/SSN1 must be set to 1 at this time.
0	SPIEN	RW	0x0	SPI enable (disabled by turning off the clock): 1: Enable SPI0 0: Disable SPI0, clear transmit and receive buffers

11.3.2 SPI Master Mode Control Register 0 (SPI_CS0) (Offset: 04H)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	TRI0_EN	RW	0x0	SPI three-wire mode enable: 1: Enable three-wire mode 0: Disable three-wire mode
6	SSN0	RW	0x0	In SPI master mode, CS0 corresponds to the SSN control in master mode. If SSNSEN is 1, software can control SSN output level by this bit: 1: SSN outputs low level 0: SSN outputs high level

Bit	Name	Attribute	Reset Value	Description
				Note: When SSNSEN = 0 (hardware auto-control), this bit must be set to 1.
5:3	BAUD0	RW	0x1	In SPI master mode, CS0 corresponds to the baud rate configuration bits (the maximum communication rate is set to 30M): 0: f_{PCLK} / 2 1: f_{PCLK} / 4 2: f_{PCLK} / 8 3: f_{PCLK} / 16 4: f_{PCLK} / 32 5: f_{PCLK} / 64 6: f_{PCLK} / 128 7: f_{PCLK} / 256 These bits shall not be modified while communication is in progress.
2	LSBF0	RW	0x0	In SPI master mode, CS0 corresponds to the frame format: 0: MSB first 1: LSB first Note: This bit shall not be modified while communication is in progress.
1	CPOL0	RW	0x0	In SPI master mode, CS0 corresponds to the clock polarity selection: 1: Serial clock idles at high level. 0: serial clock idles at low level. Note: The value of this bit cannot be modified while communication is in progress or while SSN is low.
0	CPHA0	RW	0x0	In SPI master mode, CS0 corresponds to the clock phase selection: 1: The second clock edge is the first capture edge. 0: The first clock edge is the first capture edge. Note: The value of this bit cannot be modified while communication is in progress.

11.3.3 SPI Master Mode Control Register 1 (SPI_CS1) (Offset: 08H)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	TRI1_EN	RW	0x0	SPI three-wire mode enable: 1: Enable three-wire mode 0: Disable three-wire mode
6	SSN1	RW	0x0	In SPI master mode, CS1 corresponds to the SSN control in master mode. If SSNSEN is 1, software can control SSN output level by this bit: 1: SSN outputs low level 0: SSN outputs high level Note: When SSNSEN = 0 (hardware auto-control), this bit must be set to 1.
5:3	BAUD1	RW	0x1	In SPI master mode, CS1 corresponds to the baud rate configuration bits (the maximum communication rate is set to 30M): 0: f_{PCLK} / 2 1: f_{PCLK} / 4 2: f_{PCLK} / 8 3: f_{PCLK} / 16 4: f_{PCLK} / 32 5: f_{PCLK} / 64 6: f_{PCLK} / 128 7: f_{PCLK} / 256 These bits cannot be modified while communication is in progress.
2	LSBF1	RW	0x0	In SPI master mode, CS1 corresponds to the frame format: 0: MSB first 1: LSB first Note: This bit cannot be modified while communication is in progress.
1	CPOL1	RW	0x0	In SPI master mode, CS1 corresponds to the clock polarity selection: 1: Serial clock idles at high level.

Bit	Name	Attribute	Reset Value	Description
				0: Serial clock idles at low level. Note: The value of this bit cannot be modified while communication is in progress or while SSN is low.
0	CPHA1	RW	0x0	In SPI master mode, CS1 also corresponds to the clock phase selection: 1: The second clock edge is the first capture edge. 0: The first clock edge is the first capture edge. Note: The value of this bit cannot be modified while communication is in progress.

11.3.4 SPI Process Control Register (SPI_OPCR) (Offset: 14H)

Bit	Name	Attribute	Reset Value	Description
31:6	RSV	-	-	Reserved, read as 0
5	SSNNEGC	W1C	0x0	SSN negedge clear: Writing 1 by software clears the SSN falling-edge interrupt, and writing 0 has no effect.
4	SSNPOSC	W1C	0x0	SSN posedge clear: Writing 1 by software clears the SSN rising-edge interrupt, and writing 0 has no effect.
3	TXBFC	W1C	0x0	Transmit buffer clear: Writing 1 by software clears the transmit buffer, and writing 0 has no effect.
2	RXBFC	W1C	0x0	Receive buffer clear: Writing 1 by software clears the receive buffer, and writing 0 has no effect.
1	MERRC	W1C	0x0	Master error clear: Writing 1 by software clears the SPI_IF.MERR bit.
0	SERRC	W1C	0x0	Slave error clear: Writing 1 by software clears the SPI_IF.SERR bit.

11.3.5 SPI Interrupt Control Register (SPI_IE) (Offset: 18H)

Bit	Name	Attribute	Reset Value	Description
31:11	RSV	-	-	Reserved, read as 0
10	SSNNEGIE	RW	0x0	SSN Negedge interrupt enable: 1: Enable 0: Disable
9	SSNPOSIE	RW	0x0	SSN Posedge interrupt enable: 1: Enable 0: Disable
8	RNFIE	RW	0x0	RX FIFO full interrupt enable: 1: Enable 0: Disable
7	TNFIE	RW	0x0	TX FIFO not-full interrupt enable: 1: Enable 0: Disable
6	MERRIE	RW	0x0	Master error interrupt enable: 1: Enable 0: Disable
5	SERRIE	RW	0x0	Slave error interrupt enable: 1: Enable 0: Disable
4	RXCOLIE	RW	0x0	Receive buffer overflow interrupt enable: 1: Enable 0: Disable
3	TXCOLIE	RW	0x0	Transmit buffer overflow interrupt enable: 1: Enable 0: Disable
2	IDLEIE	RW	0x0	SPI idle flag interrupt enable: 1: Enable 0: Disable
1	TXBEIE	RW	0x0	TX buffer empty interrupt enable: 1: Enable 0: Disable
0	RXBFIE	RW	0x0	Receive buffer non-empty interrupt enable: 1: Enable 0: Disable

11.3.6 SPI Interrupt Flag Register (SPI_IF) (Offset: 1CH)

Bit	Name	Attribute	Reset Value	Description
31:24	RSV	-	-	Reserved
23:20	RXFIFO_LEVEL	R	0x0	Number of data items currently stored in RX FIFO
19:16	TXFIFO_LEVEL	R	0x0	Number of data items currently stored in TX FIFO
15:11	RSV	-	-	Reserved, read as 0
10	SSNNEG	R	0x0	SSN Negedge flag: SSNNEG is set when SSN falling edge is detected.
9	SSNPOS	R	0x0	SSN Posedge flag: SSNPOS is set when SSN rising edge is detected.
8	RNF	R	0x0	SPI RX FIFO full: 1: SPI0 RX FIFO is full. 0: SPI0 RX FIFO is not full.
7	TNF	R	0x1	SPI TX FIFO not full: 1: SPI0 TX FIFO is not full. 0: SPI0 TX FIFO is full.
6	MERR	R	0x0	Master error flag: The MERR bit is set when SSN is pulled high before 8 bits are transmitted in master mode.
5	SERR	R	0x0	Slave error flag: The SERR bit is set when SSN is pulled high before 8 bits are transmitted in slave mode.
4	RXCOL	W1C	0x0	RX buffer overflow: 1: RX buffer overflows. 0: RX buffer does not overflow. Note: Software writing 1 clears this bit.
3	TXCOL	W1C	0x0	TX buffer overflow: 1: TX buffer overflows. 0: TX buffer does not overflow. Note: Software writing 1 clears this bit.

Bit	Name	Attribute	Reset Value	Description
2	IDLE	R	0x1	SPI0 idle flag, read-only: 1: SPI0 transfer being idle 0: SPI0 transfer being in progress
1	TXBE	R	0x1	TX buffer empty flag bit: 1: TX buffer empty, software writing to TXBUF clears this bit 0: TX buffer not empty
0	RXBF	R	0x0	RX buffer not-empty flag bit: 1: RX buffer not empty 0: RX buffer empty

11.3.7 SPI Transmit Buffer Register (SPI_TXBUF) (Offset: 20H)

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved, read as 0
8:0	TXBUF	W	0x0	SPI transmit buffer, transmit FIFO entry address. This IP contains a total of 8 transmit FIFOs with 9-bit data each. Write to this address to input the data to be transmitted into the FIFO.

11.3.8 SPI Receive Buffer Register (SPI_RXBUF) (Offset: 24H)

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved, read as 0
8:0	RXBUF	R	0x0	SPI receive buffer, receive FIFO entry address. This IP contains a total of 8 receive FIFOs with 9-bit data each. Read from this address to retrieve the received data from the FIFO.

11.3.9 SPI DMA Receive Setting Register (SPI_DMARXLEV) (Offset: 28H)

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved, read as 0
2:0	DMA_RX_LEV	RW	0x0	Setting of SPI0 RX FIFO DMA request: When the number of data items in RX FIFO exceeds the value set in this register, a DMA RX request is generated.

11.3.10 SPI DMA Transmit Setting Register (SPI_DMATXLEV) (Offset: 2CH)

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved, read as 0
3:0	DMA_TX_LEV	RW	0x0	Setting of SPI0 TX FIFO DMA request: When the number of data items in TX FIFO is less than the value set in this register, a DMA TX request is generated.

11.4 Operation Procedure

The two pin configurations for SPI are shown in the table below:

Table 11-2**: SPI Pin Configuration**

Signal Description Function Bit	SPI Configuration 1	SPI Configuration 2
CS	SPI_CS0	SPI_CS1
MISO	SPI_MISO	SPI_MI1
MOSI	SPI_MOSI	SPI_MOSI
CLK	SPI_SCK	SPI_SCK

Note: SPI_CS1 is used in conjunction with SPI_MI1, while SPI_CS0 is used in conjunction with SPI_MISO.

11.4.1 Initialization Process

1. Enable the SPI module clock and reset PERI_CLKEN / PERI_RESET.
2. Configure the corresponding GPIO pins to be multiplexed as SPI_SCK, SPI_MISO, SPI_MOSI and SPI_CLK, and the pins used for input functions shall be configured to enable pin input via the PADIEx register.
3. Configure SPI_CR[5] to set the master or slave mode.
4. Configure SPI_CR[10] to set the SSN control mode.
5. Configure SPI_CR[1] to set whether the SSN output is controlled by software or hardware.
6. Configure SPI_CSx[6] to set whether the SSN output is high or low.
7. Configure SPI_CSx[2] to set whether the MSB or LSB is transmitted first.
8. Configure SPI_CSx[0] to set whether the first or second clock edge is used for sampling.
9. Configure SPI_CSx[1] to set whether the serial clock stops at high level or low level.
10. Configure SPI_CSx.BAUDx[2:0] to set the serial clock baud rate (not required in slave mode, where the serial clock rate is determined by the master). Configure SPI_IE to enable the corresponding interrupts if required.
11. Configure SPI_CR[0] to enable SPI.

11.4.2 Transmission Process

- Master transmission process:

  Configure SPI_CSx[6] to pull down the SSN pin to start the transmission. Set the SPI_CR[8] bit high. Write the data to the SPI_TXBUF register. Wait for SPI_IF[2] to be set, indicating that the transmission is complete. Set the SPI_CR[8] bit low. Pull up the SSN pin after the

transmission is complete.

- Slave transmission process:

  Set the SPI_CR[8] bit high. Write the data to the SPI_TXBUF register. Wait for SPI_IF[2] to be set, indicating that the transmission is complete. Set the SPI_CR[8] bit low.

11.4.3 Reception Process

- Master reception process:

  Configure SPI_CSx[6] to pull down the SSN pin to start the transmission. Write the data to the SPI_TXBUF register. Wait for SPI_IF[0] to be set, indicating that the reception is complete. Read the data from the SPI_RXBUF register to complete the data reception. Pull up the SSN pin after the transmission is completed.

- Slave reception process:

  Wait for SPI_IF[0] to be set, and read the data from the SPI_RXBUF register to complete data reception.

11.4.4 SPI DMA Transmission Process

1. Configure SPI_DMATXLEV[2:0] to set the number of FIFO data items that trigger a DMA TX request.
2. Configure SPI_CR[13] to enable the DMA TX request.
3. Configure to enable the DMA module clock and reset PERI_CLKEN / PERI_RESET.
4. Configure the DMA channel control register DMA_CHCTRLCx, setting the data bit width and transfer mode (8-bit width, memory-to-peripheral mode) according to the actual application.

5. Configure DMA_CHCTRLCx[4:3] and DMA_CHCTRLCx[2:1] to select the destination peripheral (SPI_TX) and source peripheral (MEM). These bits take effect in peripheral mode.
6. Configure DMA_CHCTRLCx[2:1] and DMA_CHCTRLCx[4:3] to select whether the destination address and source address increment with data transfer (the source address increments while the destination address remains unchanged).
7. If interrupts are required, configure the DMA interrupt mask register DMA_INTMASK to enable the corresponding channel interrupt.
8. Configure DMA_SRCADDRCx to set the channel source address.
9. Configure DMA_DSTADDRCx to set the channel destination address.
10. Configure DMA_CHCTRLCx[29:15] to set the number of transfer blocks.
11. Wait for the above configurations as well as the corresponding source and destination addresses to be ready, then enable DMA (DMAC_EN).
12. Configure DMA_CHCTRLCx[0] to enable DMA channel transfer.
13. Wait for DMA_CHCTRLCx[0] to be 0, indicating the transfer is complete. If the transfer completion interrupt is enabled, wait for the transfer completion interrupt before proceeding.

11.4.5 SPI DMA Reception Process

1. Configure SPI_DMARXLEV[2:0] to set the number of FIFO data items that trigger a DMA RX request.
2. Configure SPI_CR[12] to enable the DMA RX request.
3. Configure to enable the DMA module clock and reset PERI_CLKEN / PERI_RESET.
4. Configure the DMA channel control register DMA_CHCTRLCx, setting the data bit width and

transfer mode (8-bit width, peripheral-to-memory mode) according to the actual application.

5. Configure DMA_CHCTRLCx[4:3] and DMA_CHCTRLCx[2:1] to select destination peripheral (MEM) and source peripheral (SPI_RX). These bits take effect in peripheral mode.
6. Configure DMA_CHCTRLCx[2:1] and DMA_CHCTRLCx[4:3] to select whether the destination address and source address increment with data transfer (the source address remains unchanged while the destination address increments).
7. If interrupts are required, configure the DMA interrupt mask register DMA_INTMASK to enable the corresponding channel interrupt.
8. Configure DMA_SRCADDRCx to set the channel source address.
9. Configure DMA_DSTADDRCx to set the channel destination address.
10. Configure DMA_CHCTRLCx[29:15] to set the number of transfer blocks.
11. Wait for the above configurations as well as the corresponding source and destination addresses to be ready, then enable DMA (DMAC_EN).
12. Configure DMA_CHCTRLCx[0] to enable DMA channel transfer.
13. Wait for DMA_CHCTRLCx[0] to be 0, indicating the transfer is complete. If the transfer completion interrupt is enabled, wait for the transfer completion interrupt before proceeding.

12 ATIMER

12.1 Overview

The chip includes an advanced timer. The advanced timer consists of a 16-bit auto-reload counter and a programmable prescaler. It may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare, PWM, and complementary PWM with dead-time insertion).

12.2 Main Features

- 16-bit up, down, up/down auto-reload counter
- 16-bit programmable prescaler allowing real-time adjustment of the counter clock frequency division
- 4 independent channels available for input capture, output compare, PWM, and single-pulse output
- Complementary output with programmable dead-time insertion
- Supporting cascading between timers
- Repetition counter, supporting timer state update after multiple cycles
- 7 break input pins, comparator break, SVD break, break signal filtering and polarity selection, and combinatorial configuration of break signals
- Interrupt or DMA events can be generated on the following occurrences:
    - Counter overflow/underflow, counter initialization (software or hardware trigger)
    - Trigger events (counter start, stop, initialization, internal/external trigger)

- Input capture
- Output compare
- Break signal input

- Supporting incremental quadrature encoder and Hall sensor

12.3 ATIMER Block Diagram

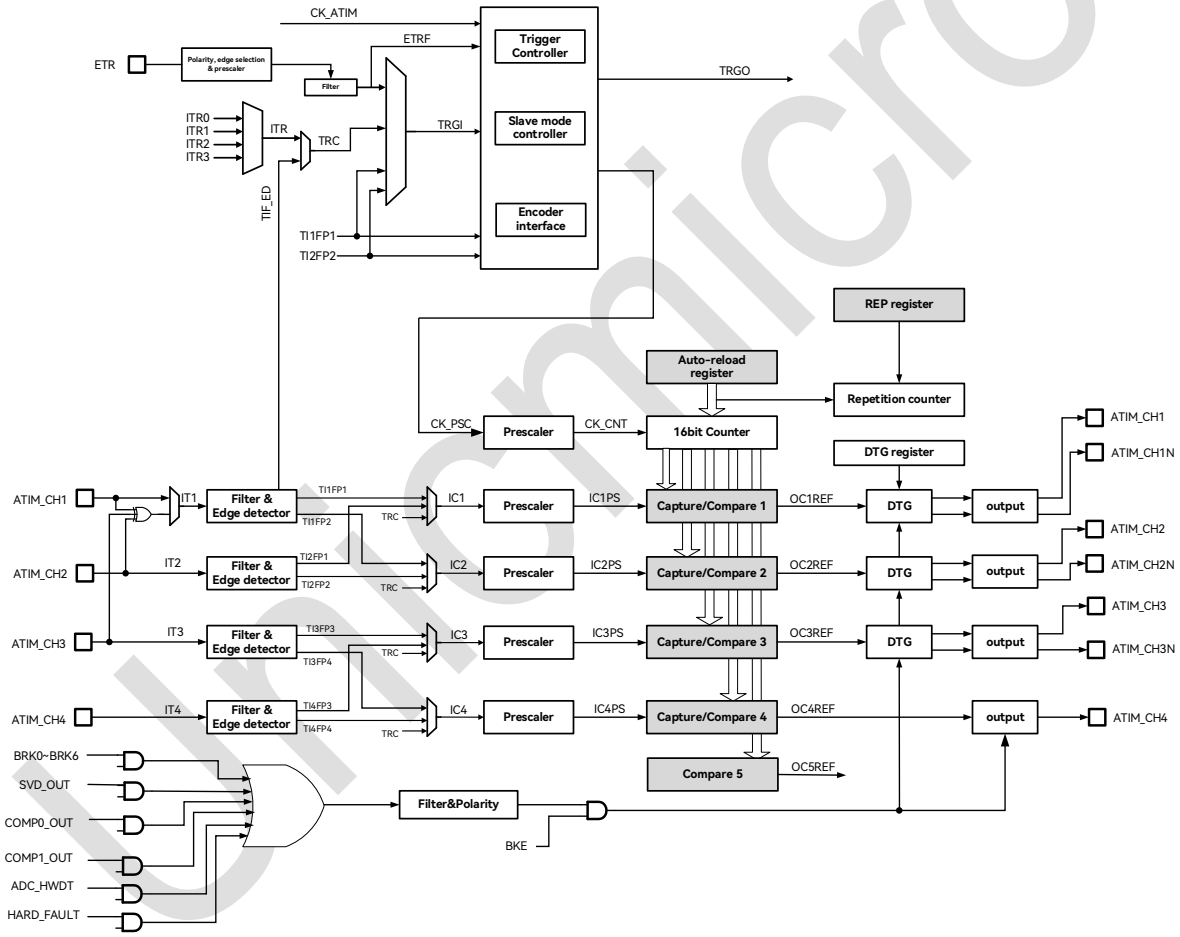


Figure 12-1: ATIMER Block Diagram

12.4 Functional Description

12.4.1 Timing Unit

The timing unit consists of a 16-bit counter and an auto-reload register. The counter can count up, down, or both up and down. The counter clock is derived from the clock after being divided by a 16-bit prescaler.

The counter, auto-reload register and prescaler register can all be rewritten or read by software, even while the counter is running.

The timing unit includes the following registers:

- Counter (ATIMER_CNT)
- Prescaler register (ATIMER_PSC)
- Auto-reload register (ATIMER_ARR)
- Repetition counter register (ATIMER_RCR)

The ATIMER_ARR features a preload function, which is controlled by the ARPE (auto-reload preload enable) register. When ARPE = 0, writing to the ATIMER_ARR register directly transfers the data to the shadow register. When ARPE = 1, data written to the ATIMER_ARR register is transferred to the shadow register upon an update event (ATIMER_CNT overflow or underflow). Software can also actively trigger an ATIMER_ARR update (UEV) through register operations.

The working clock of ATIMER_CNT is driven by the clock divided by ATIMER_PSC. The ATIMER_CNT only starts counting when the counter enable register (CEN) is set. When ATIMER_CNT = ATIMER_ARR, the current counting cycle ends, and an update event is triggered.

ATIMER_PSC is a synchronous prescaler that can divide the clock by 1 to 65536. The ATIMER_PSC register is also buffered. Writing to ATIMER_PSC does not directly update the

shadow register. Instead, the new prescaler value is updated to the shadow register only upon the next update event. Therefore, during the counting process of ATIMER_CNT, software can dynamically rewrite ATIMER_PSC, and the new prescaler ratio will be adopted at the next update event.

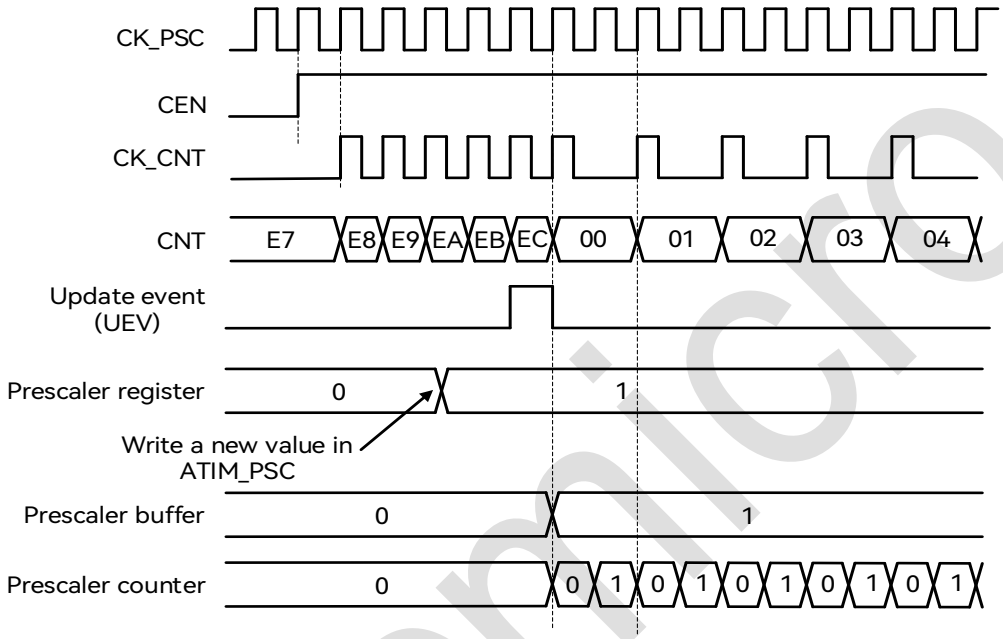


Figure 12-2: Counter Timing Diagram with Prescaler Division Changing from 1 to 2

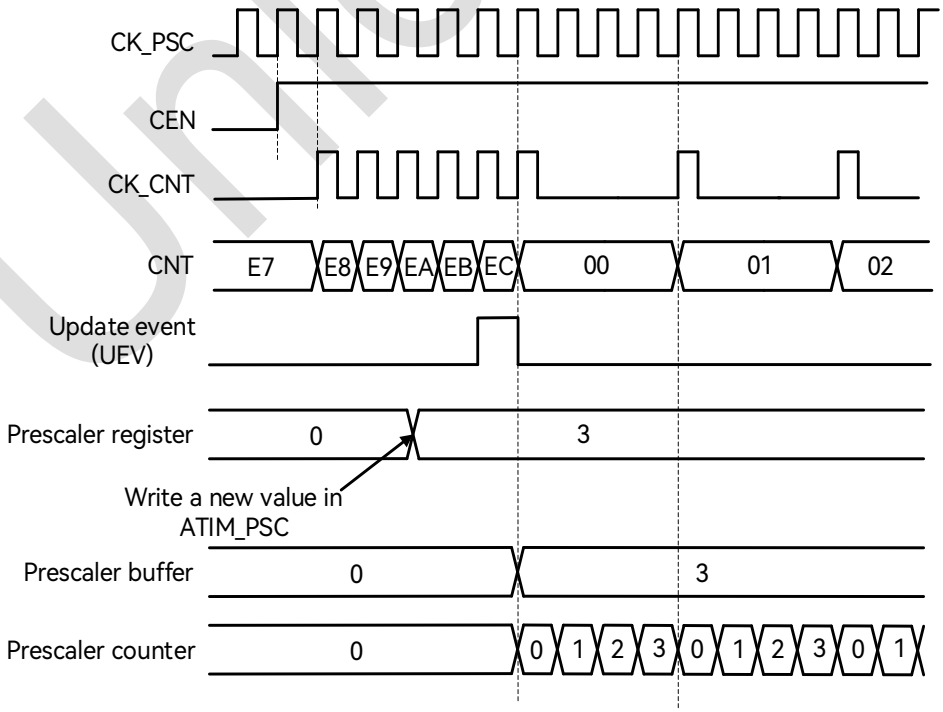


Figure 12-3: Counter Timing Diagram with Prescaler Division Changing from 1 to 4

12.4.2 Counter Operation Mode

The counter supports up-counting mode, down-counting mode and center-aligned mode.

12.4.3 Up-counting Mode

In up-counting mode, the counter counts from 0 until ATIMER_CNT = ATIMER_ARR, generating an overflow event, and then restarts from 0.

If the repetition counter is enabled, the counter repeats the above process a number of times (ATIMER_RCR + 1) defined by ATIMER_RCR before generating an overflow event.

Software can directly trigger an update event by setting the UG register, at which time the ATIMER_CNT and the prescaler register are automatically cleared. Whether setting the UG register triggers UIF (update interrupt flag) to be set is determined by the setting of the URS register.

The update event can be disabled by setting the UDIS register to prevent the values in the preload registers from being updated to the working registers.

When an update event occurs, the following registers are updated and the UIF is set:

- The shadow register of ATIMER_RCR is updated with the content of ATIMER_RCR.
- The shadow register of ATIMER_ARR is updated with the content of ATIMER_ARR.
- The shadow register of ATIMER_PSC is updated with the content of ATIMER_PSC.

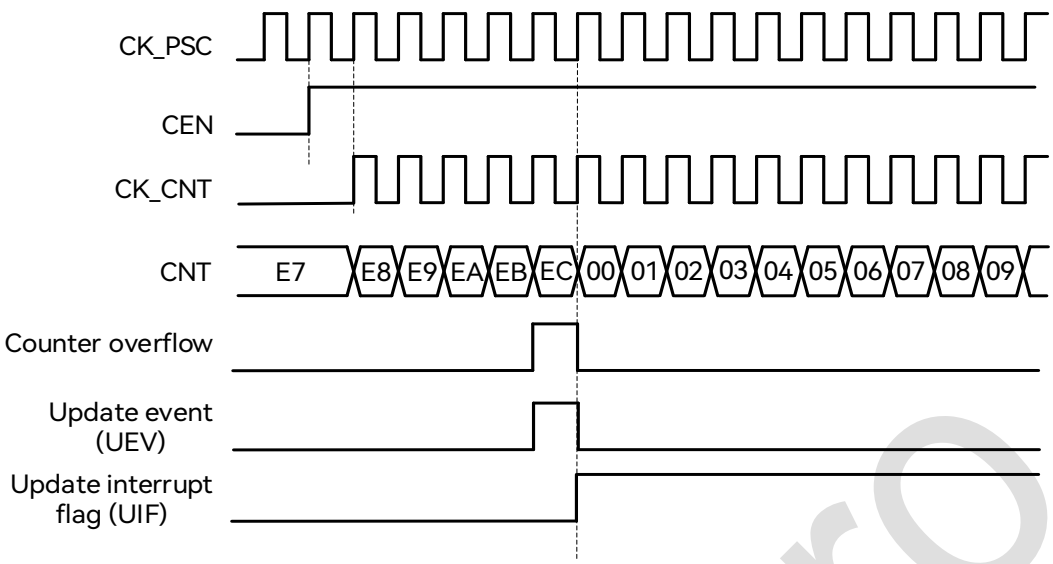


Figure 12-4: Up-counting Waveform, Internal Clock not Divided

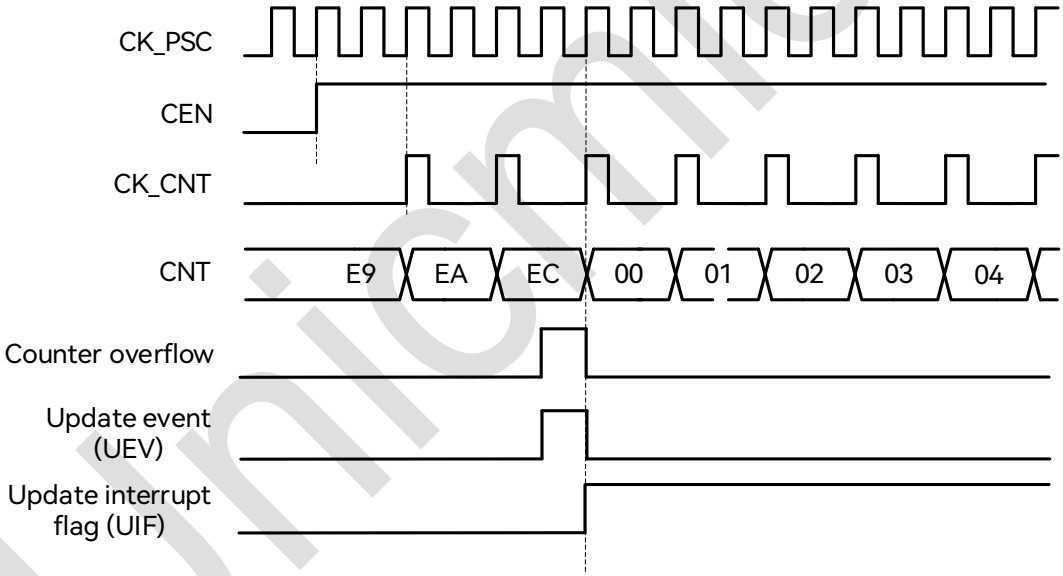


Figure 12-5: Up-counting Waveform, Internal Clock Divided by 2

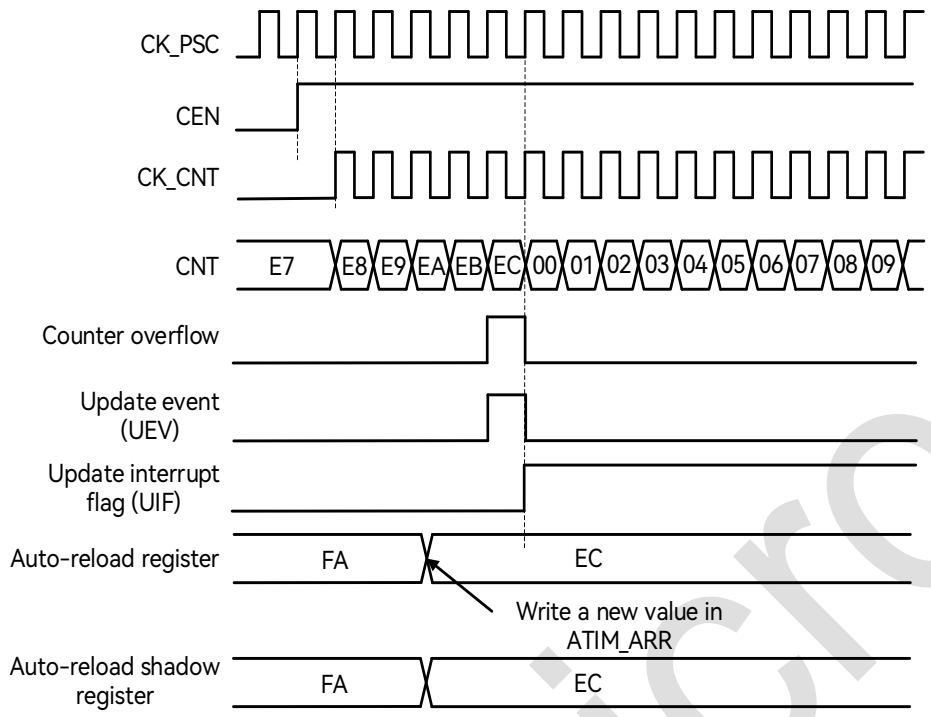


Figure 12-6: Update Event when ARPE = 0 (ATIMER_ARR not Preloaded)

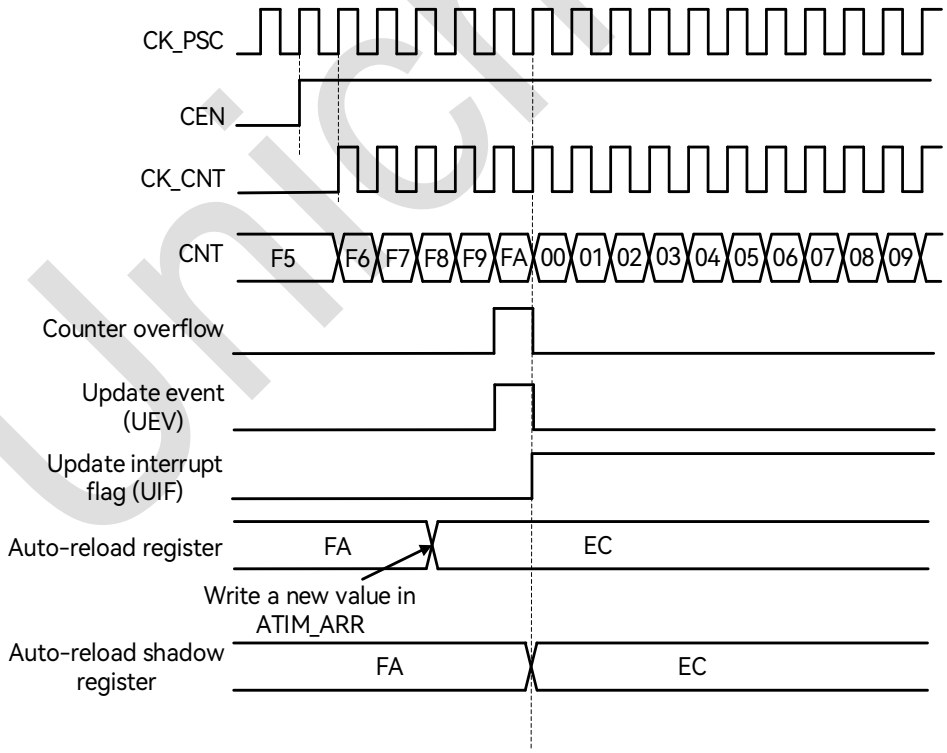


Figure 12-7: Update Event when ARPE = 1 (ATIMER_ARR Preloaded)

12.4.4 Down-counting Mode

In down-counting mode, the counter counts from the ATIMER_ARR value down to 0, generating an underflow event, and then restarts counting from the ATIMER_ARR value.

If the repeat counter is enabled, the counter repeats the above process a number of times (ATIMER_RCR + 1) defined by ATIMER_RCR before generating an underflow event.

Software can directly trigger an update event by setting the UG register, at which time the ATIMER_CNT and the prescaler register are automatically cleared. Whether setting the UG register triggers UIF (update interrupt flag) to be set is determined by the setting of the URS register.

The update event can be disabled by setting the UDIS register to prevent the values in the preload registers from being updated to the working registers.

When an update event occurs, the following registers are updated and the UIF is set:

- The shadow register of ATIMER_RCR is updated with the content of ATIMER_RCR.
- The shadow register of ATIMER_ARR is updated with the content of ATIMER_ARR.
- The shadow register of ATIMER_PSC is updated with the content of ATIMER_PSC.

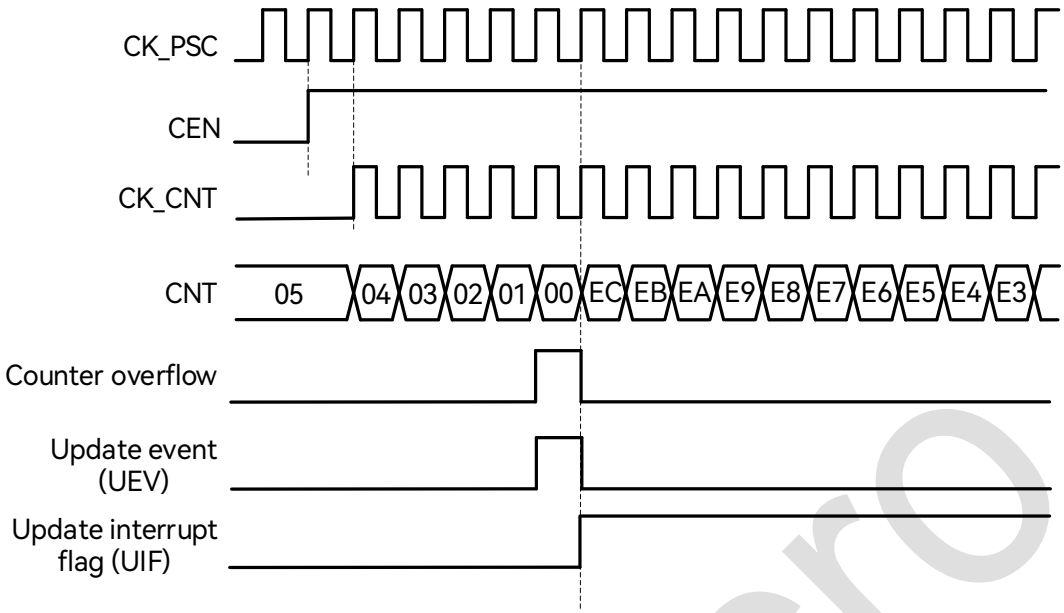


Figure 12-8: Down-counting Waveform, Internal Clock not Divided

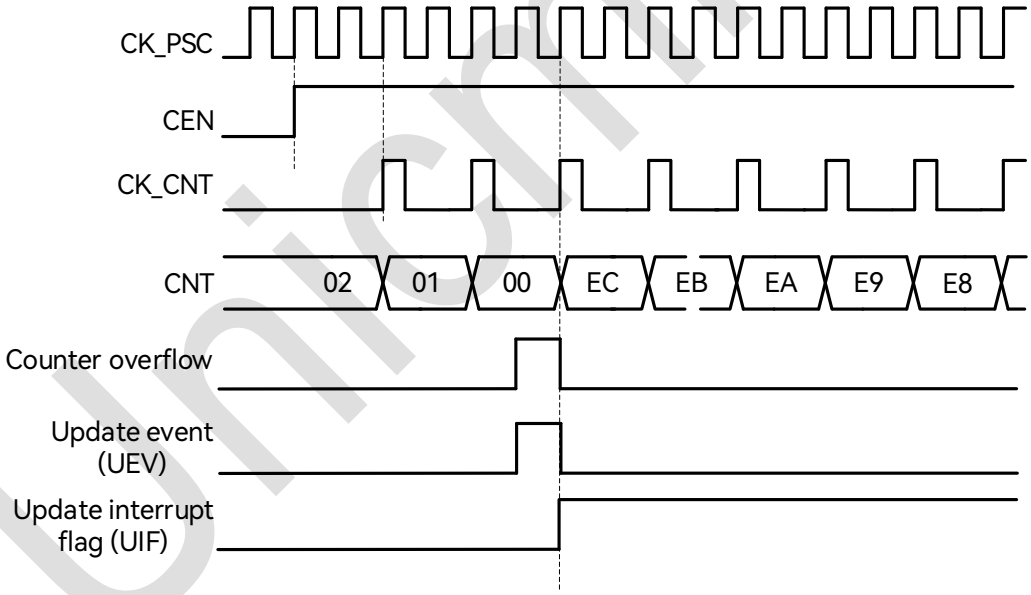


Figure 12-9: Down-counting Waveform, Internal Clock Divided by 2

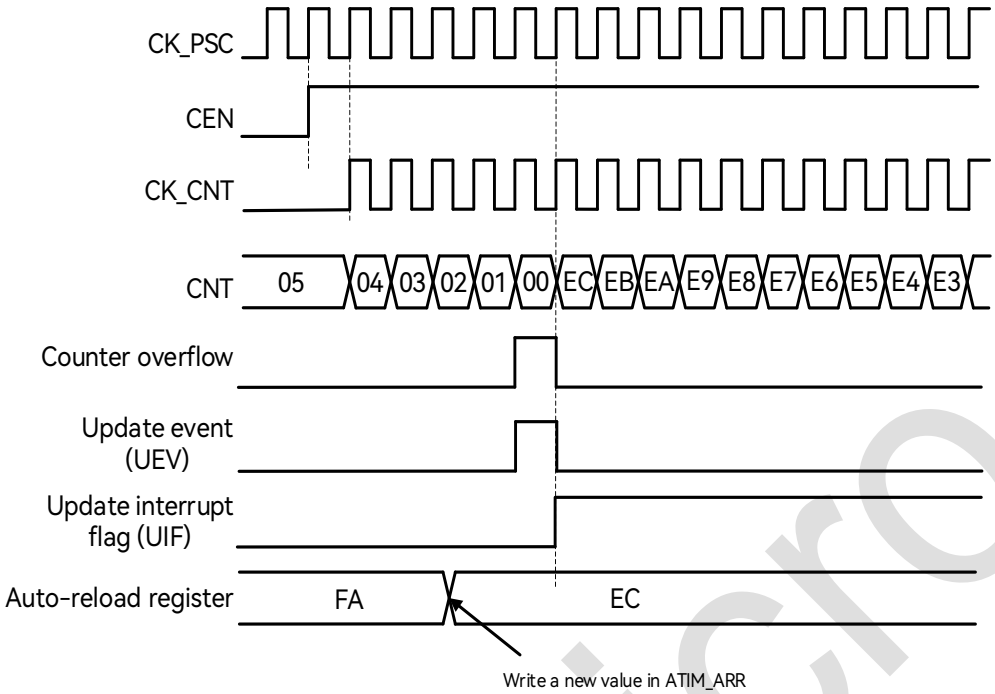


Figure 12-10: Update Event when ARPE = 0 (ATIMER_ARR not Preloaded)

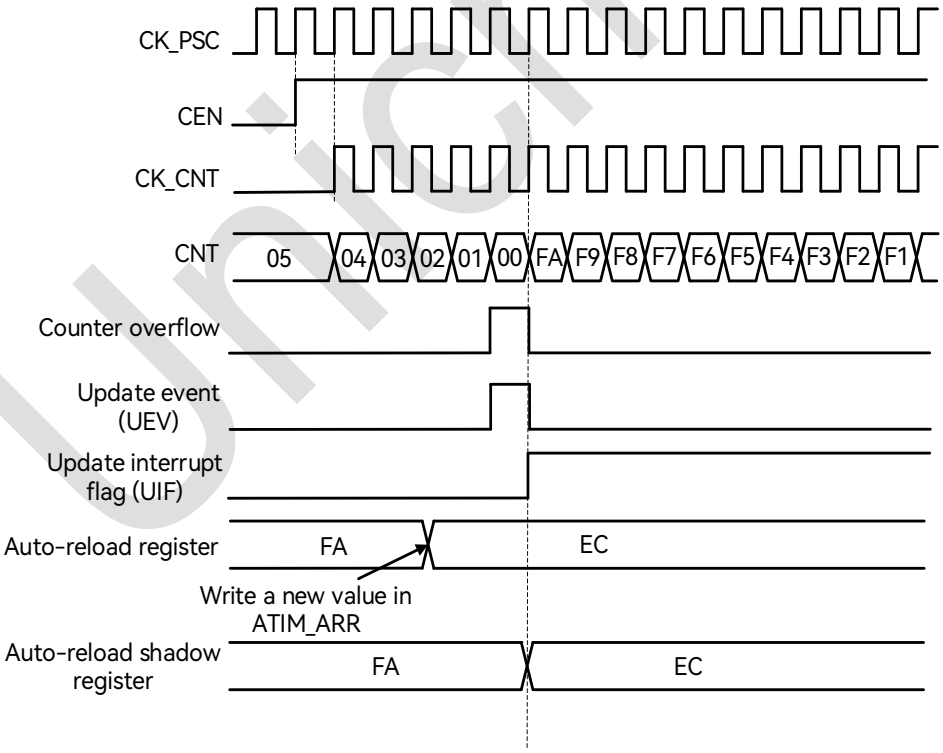


Figure 12-11: Update Event when ARPE = 1 (ATIMER_ARR Preloaded)

12.4.5 Center-aligned Counting Mode

In the center-aligned mode, the counter starts counting up from 0 to ATIMER_ARR-1, generating an overflow event, then starts counting down from ATIMER_ARR to 1, generating an underflow event, and then restarts counting up from 0.

The CMS[1:0] bits in the register are used for enabling the center-aligned mode and selecting the output compare mode herein. When CMS! = 00, the counting is center-aligned. The output compare function is only valid during down-counting when CMS = 01, during up-counting when CMS = 10, and during both up-counting and down-counting when CMS = 11.

In this mode, the DIR register cannot be rewritten by software, but is automatically updated by hardware according to the counting direction, indicating the current counting direction.

The shadow registers of ATIMER_ARR, ATIMER_PSC and ATIMER_RCR are updated on both overflow and underflow events.

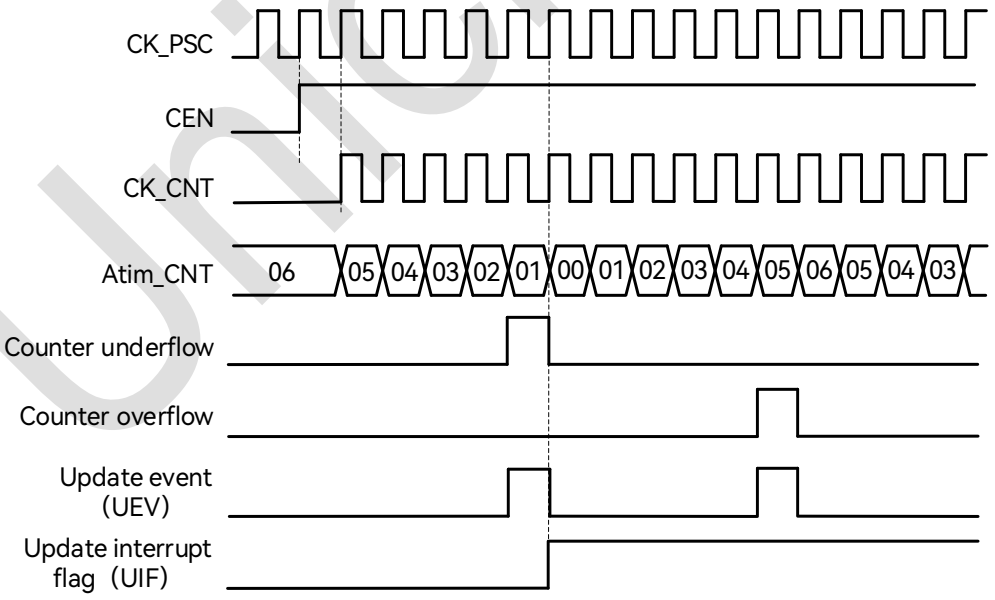


Figure 12-12: Center-aligned Counter Timing Diagram, ATIMER_PSC = 0, ATIMER_ARR = 0x6

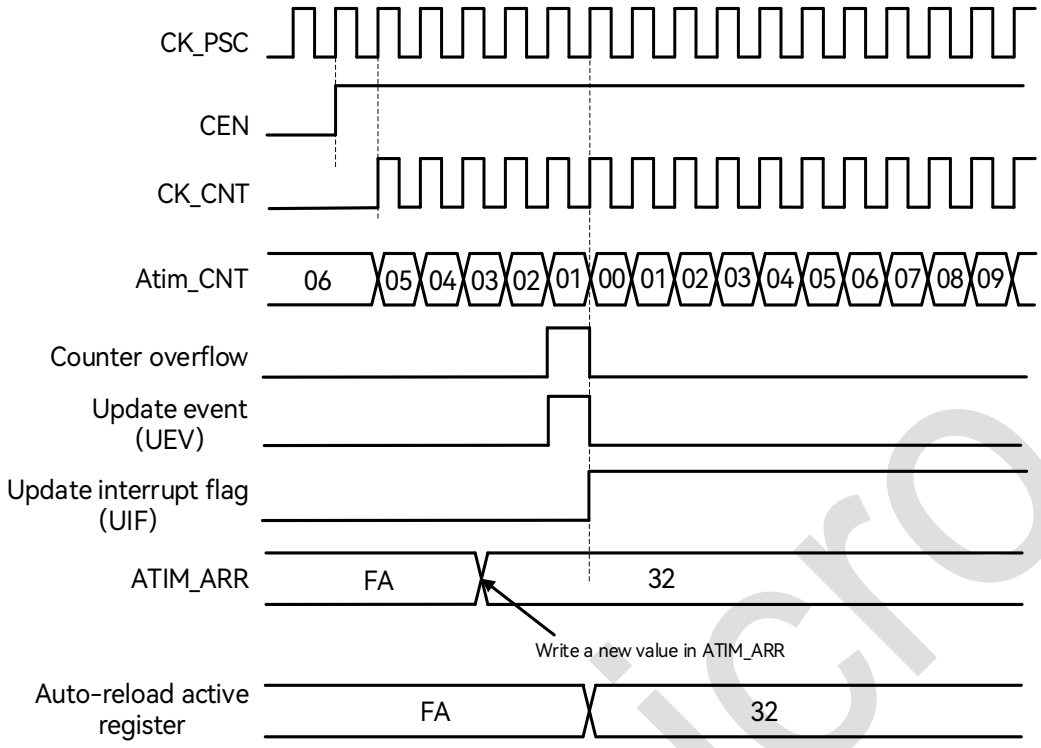


Figure 12-13: Counter Timing Diagram, Update Event When ARPE = 1 (Counter Underflow)

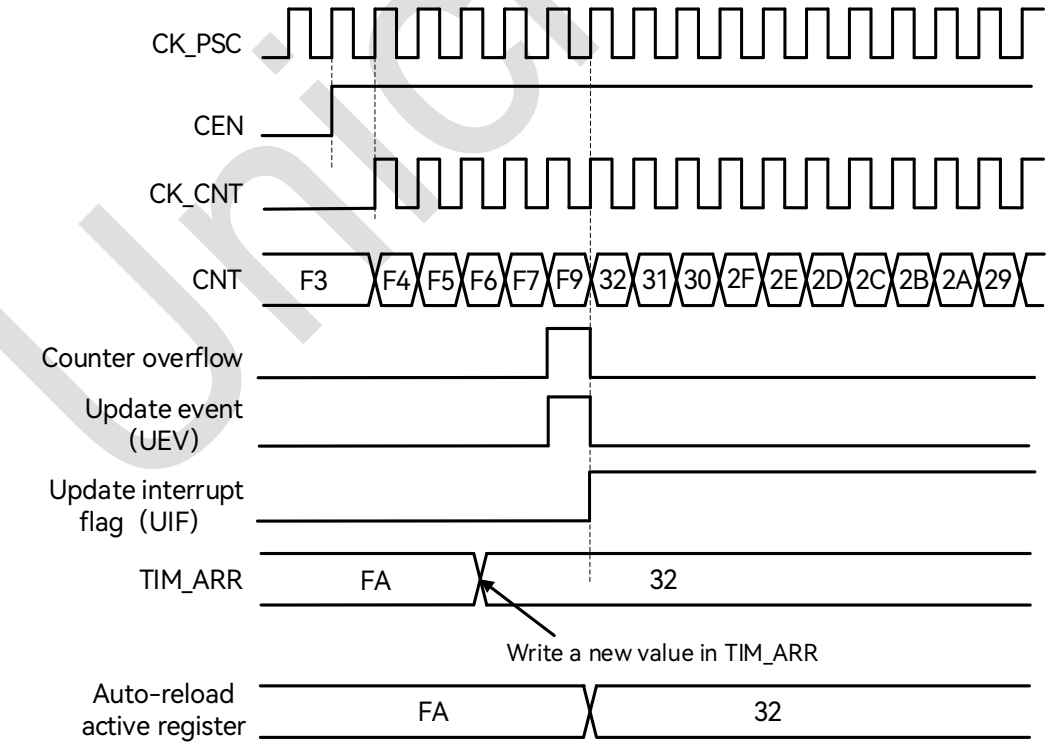


Figure 12-14: Counter Timing Diagram, Update Event When ARPE = 1 (Counter Overflow)

12.4.6 Repetition Counter

The update event occurs in the case of a counter overflow or underflow, and when the repetition counter is zero. This means that the data are transferred from the preload registers of ATIMER_ARR, ATIMER_PSC and ATIMER_CCR (compare/capture register, in output compare mode) to the shadow registers after N + 1 counter overflows/underflows, where N is the value of the ATIMER_RCR register.

The repetition counter decrements under the following conditions:

- An overflow occurs in up-counting mode.
- An underflow occurs in down-counting mode.
- Every overflow or underflow in center-aligned counting mode.

Note: When the update event is triggered by software or the slave mode controller, the update event occurs immediately, regardless of the current value of ATIMER_RCR, and the repetition counter is also immediately updated to the value of ATIMER_RCR.

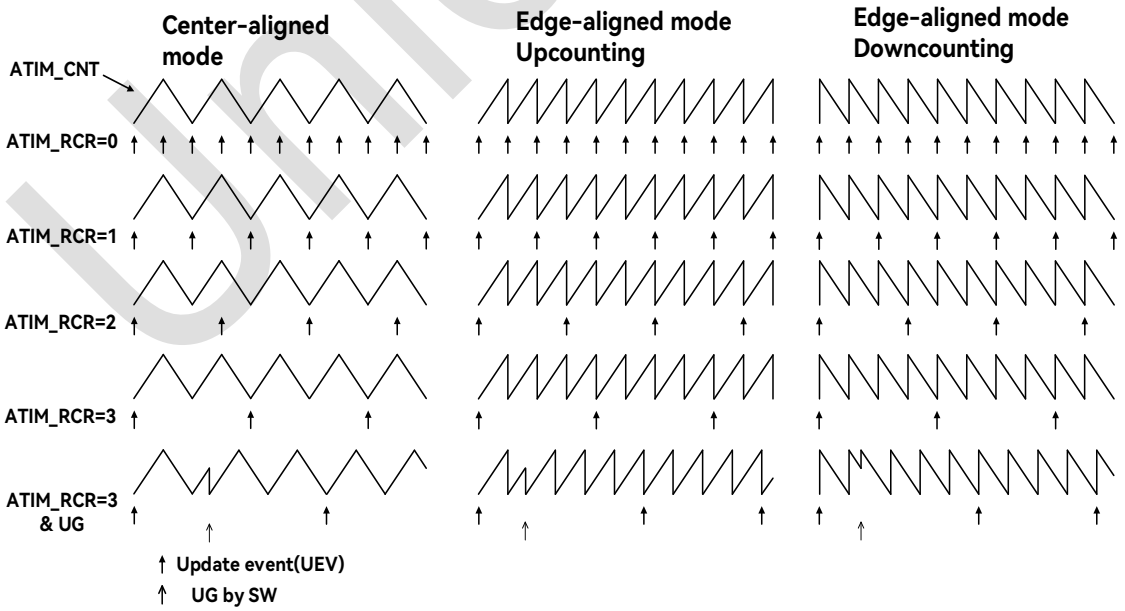


Figure 12-15: Examples of Update Rates in Different Modes and the Setting of ATIMER_RCR Register

12.4.7 Preload Register

The following functional registers support the preload function:

- Auto-reload register ATIMER_ARR
- Prescaler register ATIMER_PSC (preload function cannot be disabled)
- Channel control register ATIMER_CCR
- CCxE and CCxNE control registers
- OCxM control register

The preload function can be enabled or disabled by software for all of the above registers except ATIMER_PSC.

Registers with preload function contains two physical entities:

- Shadow register: the register actually used by the timer
- Preload register: the register accessible by software

When the preload function is disabled, the register with preload function has the following characteristics:

- The preload register can be accessed and overwritten by software in real time.
- The shadow register is updated synchronously with the preload register.

If preload function is enabled, then:

- All software operations access the preload register.
- At the occurrence of update event, the contents of all preload registers will be synchronously transferred to the corresponding shadow registers.

12.4.8 Counter Operating Clock

The counter can operate using the following clocks:

- APBCLK (CK_INT): Internal clock mode
- External pin input clock (TIx): External clock mode 1
- External pin trigger input (ETR): External clock mode 2
- Internal trigger (ITRx): Using the trigger output (TRGO) from a timer as the counting clock

12.4.9 Internal Clock Mode

In the internal clock mode, the slave mode is disabled (SMS = 000), and the CEN, DIR, UG, and other register bits are controlled by software. After software operates the UG register, the update signal is synchronized by CLK_PSC, and then the counter value will be reinitialized.

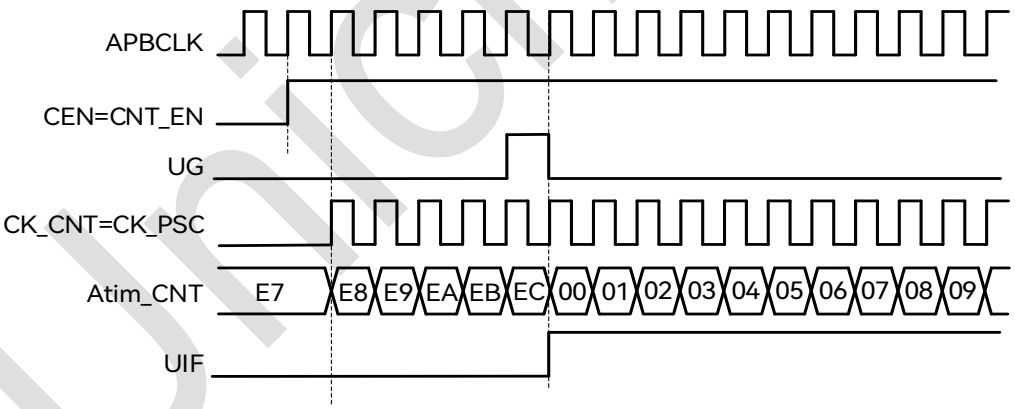


Figure 12-16: Internal Clock Source Mode, Clock Division Factor being 1

12.4.10 External Clock Mode 1

In this mode, the external pin input signal is directly used as the counting clock, with SMS configured to 111, and the counting edge can be configured as either rising or falling edge.

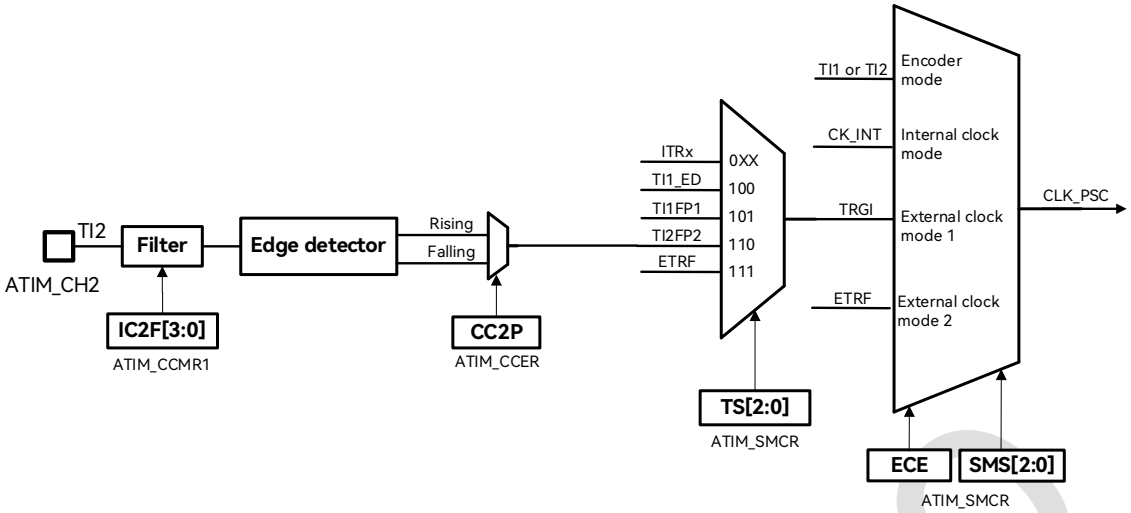


Figure 12-17: Example of External Clock Connection

The external input signal will be synchronized with the internal clock before triggering the counter counting. Meanwhile, the TIF flag will be triggered by the valid edge of the input signal.

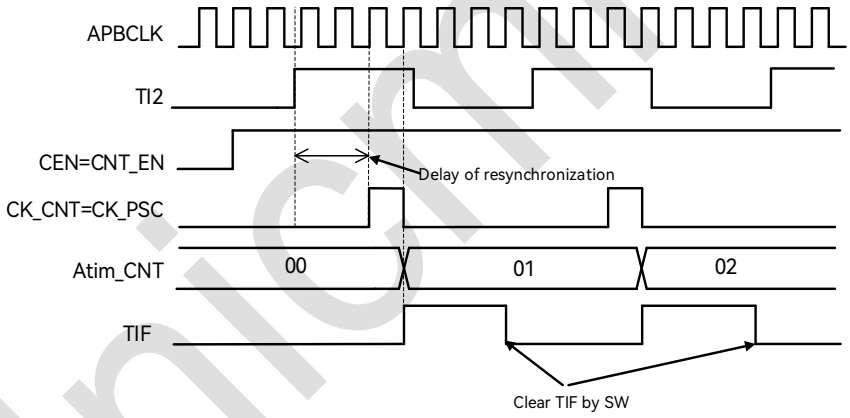


Figure 12-18: Timing Diagram for External Clock Mode 1

When counting with an external clock, the internal clock of ATIMER (ATIMERx_clk) shall still be enabled so that ATIMER can use ATIMERx_clk to synchronize and filter the external input clock. In external clock mode 1, the external input clock is first subject to filtering and edge selection to obtain the valid counting edge, which serves as the valid operating clock (CLK_PSC) input to the prescaler module.

The external clock synchronization adopts a simple two-stage flip-flop structure. To avoid metastability, the external input clock width is required to be at least two APBCLK cycles.

In this mode, only the inputs from channels 1 and 2 can be used as clock inputs, and the required configurations are as follows:

1. In GPIO module, configure the corresponding pin as ATIMER_CH2.
2. Disable the channel, setting ATIMER_CCER.CC2E = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting ATIMER_CCMR1.CC2S = 01, with IC2 mapped on TI2.
4. Select the active counting edge to be rising edge or falling edge by setting ATIMER_CCER.CC2P = 0.
5. Configure the input filtering time by setting ATIMER_CCMR1.IC2F[3:0] (IC2F = 0000, no input filtering).
6. Enable the external clock mode 1 by setting ATIMER_SMCR.SMCR = 111.
7. Select the trigger input source by setting ATIMER_SMCR.TS = 110, designating TI2 as the trigger input source.
8. Enable the channel by setting ATIMER_CCER.CC2E = 1.
9. Enable the counter by setting ATIMER_CR1.CEN = 1.

The following diagram is a typical example of external clock mode 1:

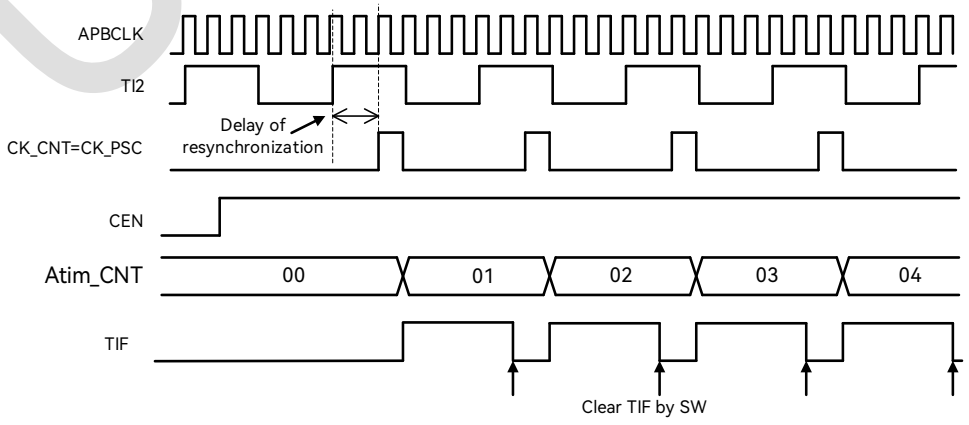


Figure 12-19: Timing Diagram for External Clock Mode 1

12.4.11 External Clock Mode 2

In this mode, the rising or falling edge (not both edges) of the signal input on the ATIMER_ETR pin is used for counting.

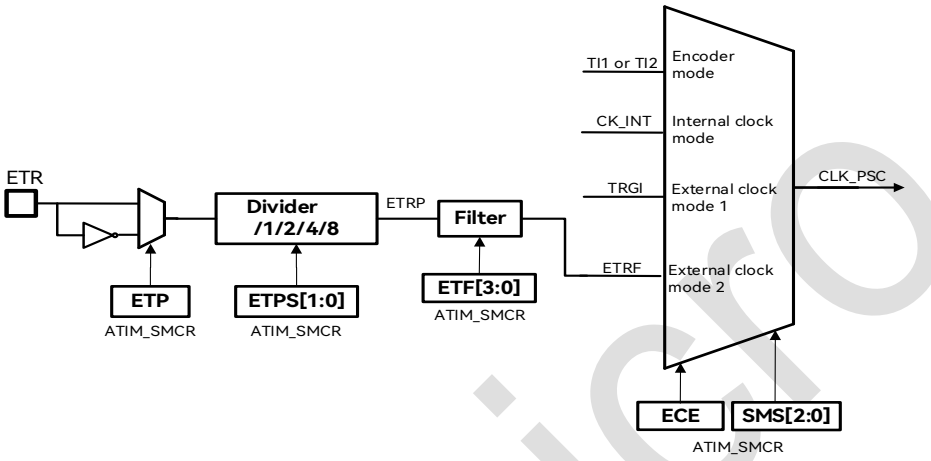


Figure 12-20: Block Diagram of External Trigger Input

The following diagram illustrates counting based on the rising edge of the ETR signal after a divide-by-two operation, where the actual counting time is delayed compared to the rising edge of the ETR input due to internal clock synchronization.

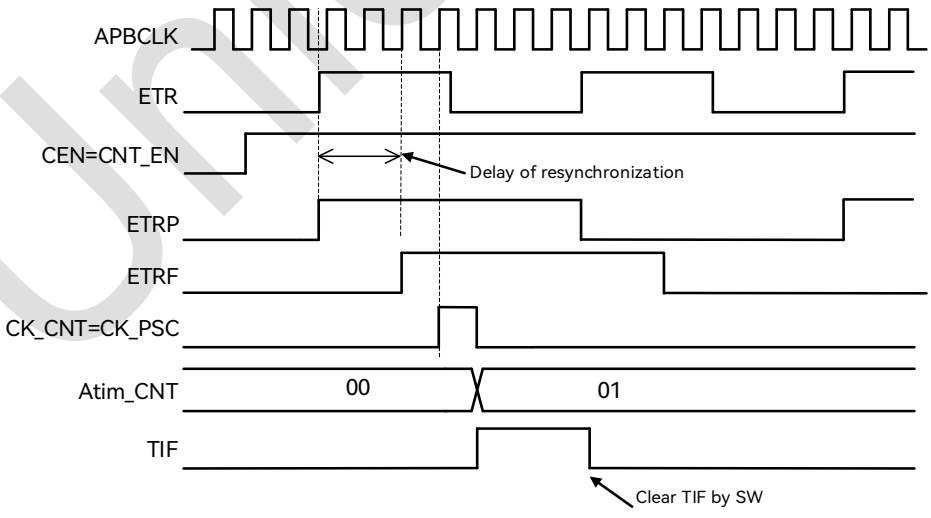


Figure 12-21: Timing Diagram 1 for External Clock Mode 2

The main difference from external clock mode 1 is that the ETR input is directly divided and then filtered to generate CK_PSC clock, which means that the applications with ETR input frequency

higher than APBCLK are supported. In such cases, the ETR input shall be pre-divided first before it is used to drive the counter.

The configuration required for this mode is as follows:

1. In GPIO module, configure the corresponding pin for ATIMER_ETR function.
2. Configure ETP for edge selection by setting ATIMER_SMCR.ETP = 0.
3. Configure the ETR prescaler ratio by setting TIM_SMCR.ETPS[1:0] = 01.
4. Configure the input filter time by setting ATIMER_SMCR.ETF[3:0] = 0000.
5. Set the ECE register and enable the external clock mode 2 by setting ATIMER_SMCR.ECE = 1 and ATIMER_SMCR.SMS = 000.
6. Enable the counter by setting ATIMER_CR1.CEN = 1.

The following diagram is a typical example of external clock mode 2:

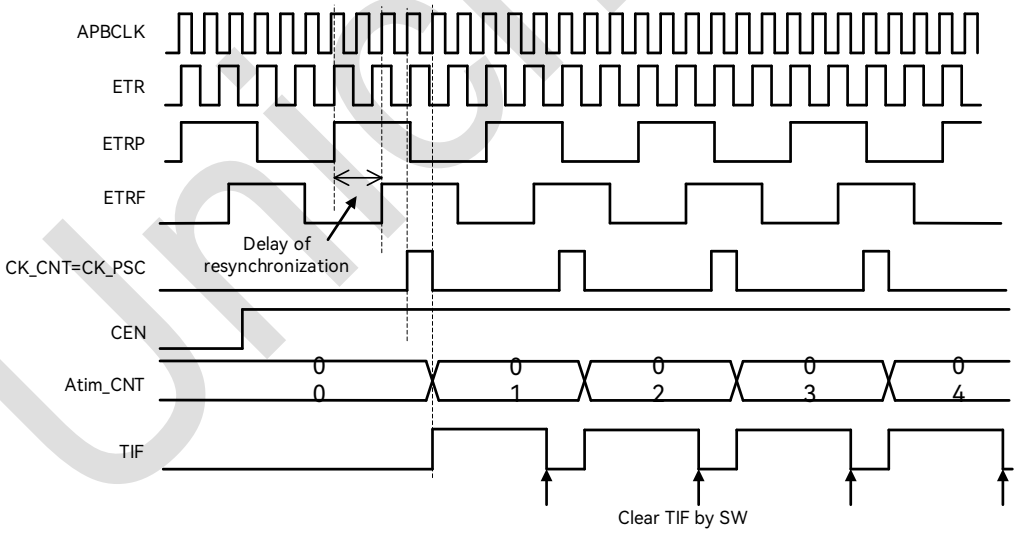


Figure 12-22: Timing Diagram for External Clock Mode 2

In external clock mode 2, ATIMER can still be configured in slave mode: for example, ETR input is used for counting while TRGO of another timer is used as the trigger signal. When a trigger event occurs, the counter can be reset and start counting again.

12.4.12 Internal Trigger Signal (ITRx)

ATIMER supports four internal trigger inputs, which can be used for counting triggers or internal signal capture. For internal signal capture, it is required to configure TS as 000–011 to select ITR0–ITR3, and configure CCxS as 11 to select TRC as the capture signal.

ITR inputs support four internal signal extensions, configured by the TS register. Refer to the following table for input signal sources:

Table 12-1: Input Signal Sources

Slave	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
ATIMER	GTIMER1_TRGO	GTIMER2_TRGO	COMP0_IN	COMP1_IN

12.4.13 Capture/Compare Channels

The ATIMER contains of 4 capture channels and 5 compare channels. Each channel consists of a capture/compare register (CCR) with a shadow register, a capture input stage, and a compare output stage.

The input stage circuit samples the TIx input and generates a filtered signal TIxF, which is then processed to generate the corresponding TIxFPx signal through edge detection and polarity selection. This signal can serve as a counting trigger or a signal to be captured, and it is prescaled before being captured.

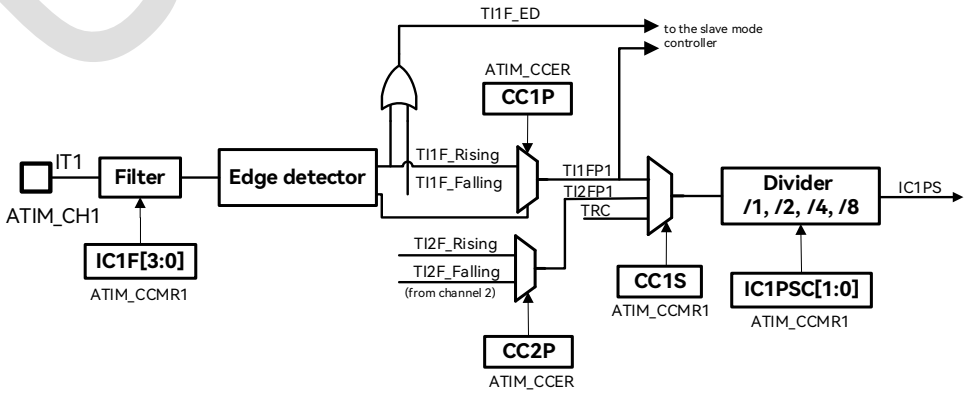


Figure 12-23: Capture/Compare Channel (Input Section of Channel 1)

The output stage circuit generates an output reference signal OCxREF, which is always active high and serves as the reference input for the final output circuit. Channels 1–3 support complementary output and dead-time insertion, while channel 4 is relatively simple and does not support complementary output.

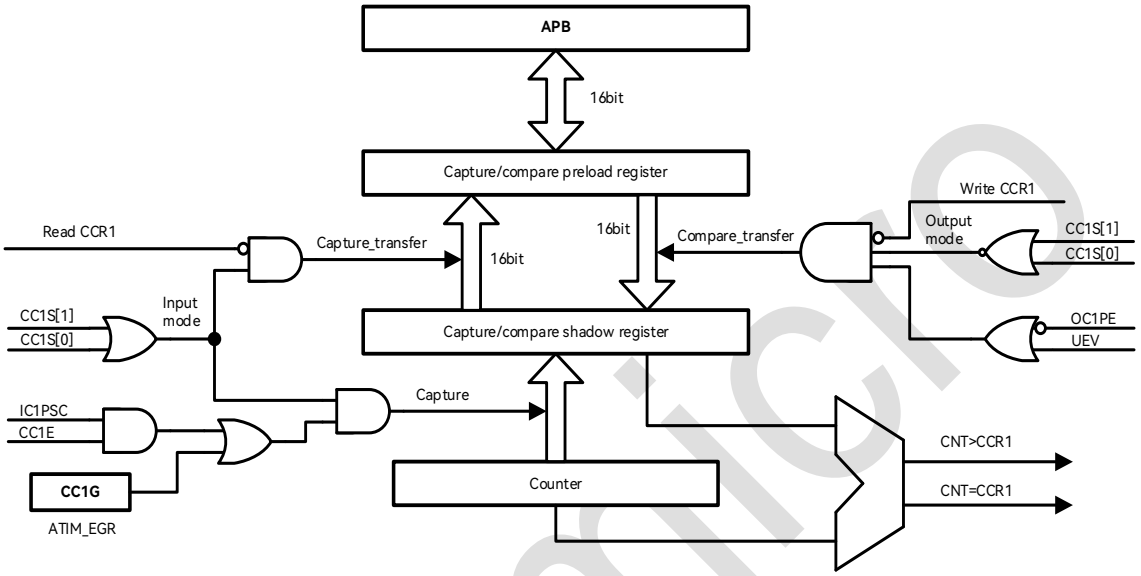


Figure 12-24: Main Circuit of Capture/Compare Channel 1

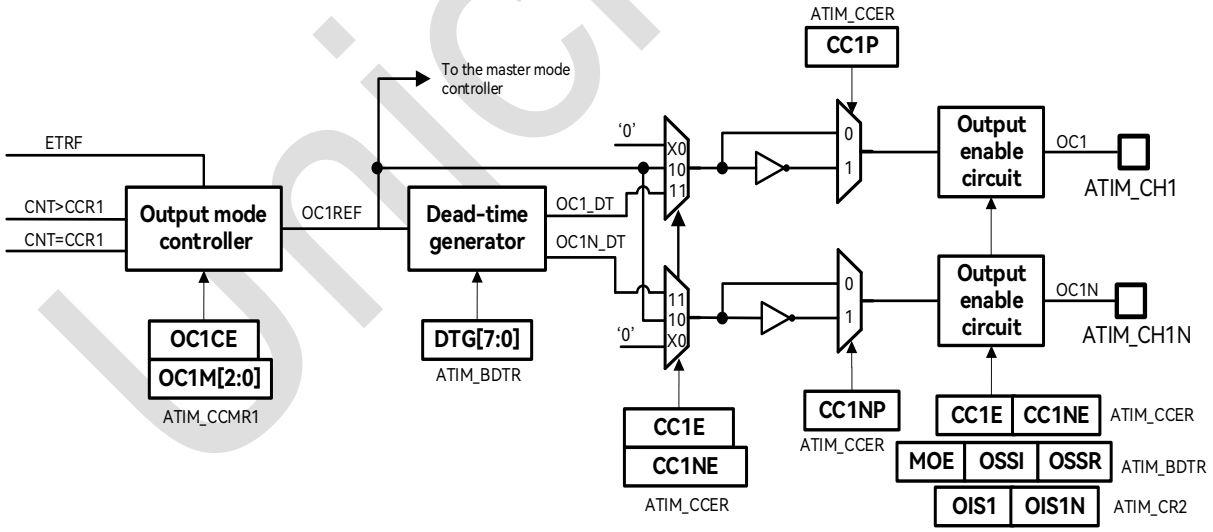


Figure 12-25: Output Section of Capture/Compare Channels (Channels 1–3)

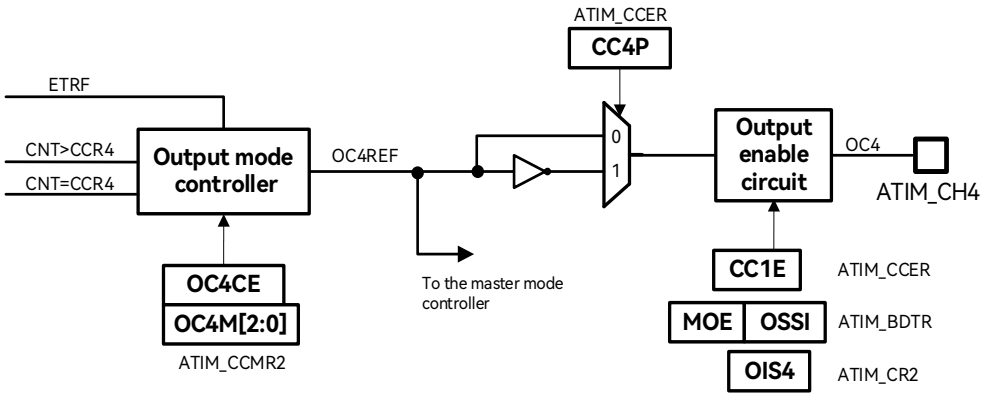


Figure 12-26: Output Section of Capture/Compare Channel 4

The capture/compare register (CCR) includes a preload register and a shadow register. Software read and write operations always access the preload register. In capture mode, the captured value is stored in the shadow register and then copied to the preload register. In compare mode, the value in the preload register is copied to the shadow register for comparison with the counter.

12.4.14 Input Capture Mode

When the expected level transition occurs on the ICx signal, a capture is triggered, and the current counter value is latched into the ATIMER_CCR. At the same time, the CCxIF interrupt flag is set, and the corresponding interrupt or DMA request can be triggered. If a capture event occurs while CCxIF is high, the capture data conflict flag (CCxOF, over-capture) is set (the last captured value in CCR is overwritten). CCxIF can be cleared by software or automatically cleared by reading the ATIMER_CCR register. CCxOF can be cleared by software writing 1 to it.

The input capture of PWM signals can be realized through the cooperation of two or more channels. For example, to calculate the period and duty cycle of an input signal, the signal can be input from the TI1 pin. The chip internally takes the rising edge of the filtered signal to get TI1FP1 and the falling edge to get TI1FP2. TI1FP1 is input to capture channel 1, and TI1FP2 is input to capture channel 2, so that channel 1 captures the rising edge of the input signal and channel 2 captures the falling edge. After the capture interrupt occurs periodically, the software

can calculate the period and duty cycle of the input signal through the values of ATIMER_CCR1 and ATIMER_CCR2 registers.

To capture the counter value on the rising edge of the input signal on TI1 to the ATIMER_CCR1 register, the configuration steps are as follows:

1. In GPIO module, configure the corresponding pin as ATIMER_CH1.
2. Disable the channel by setting ATIMER_CCER.CC1E = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting ATIMER_CCMR1.CC1S = 01, mapping IC1 to TI1.
4. Select the valid counting edge to be rising edge or falling edge by setting ATIMER_CCER.CC1P.
5. Configure the input filter time by setting ATIMER_CCMR1.IC1F[3:0].
6. Configure the input prescaler by setting ATIMER_CCMR1.IC1PS[1:0].
7. Enable the channel by setting ATIMER_CCER.CC1E = 1.

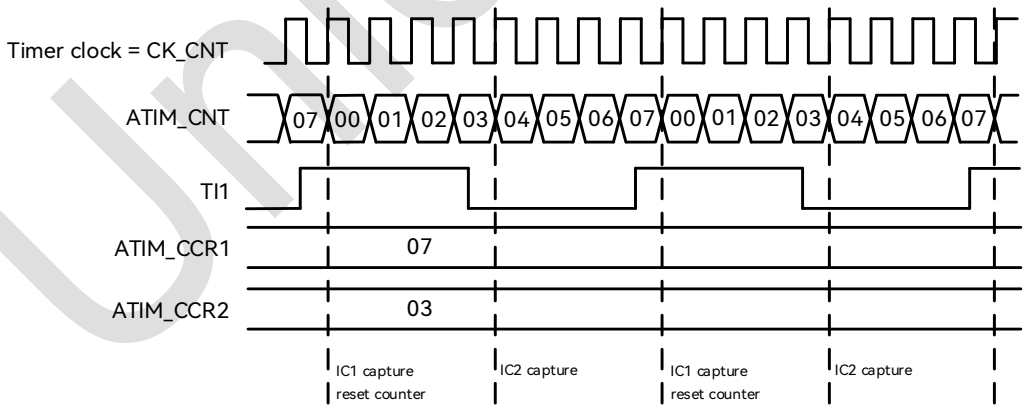


Figure 12-27: PWM Input Capture Mode Timing Diagram

To implement the PWM input capture function, the following settings are required:

1. In GPIO module, configure the corresponding pin as ATIMER_CH1.
2. Disable the channel by setting ATIMER_CCER.CC1E = 0 and ATIMER_CCER.CC2E = 0 to

ensure the success of subsequent channel configuration.

3. Select the input channels, where both channels IC1 and IC2 are mapped to the same TI1 input port, by setting ATIMER_CCMR1.CC1S = 01 and ATIMER_CCMR1.CC2S = 10.
4. Select the counting edge, where the polarity of the valid edges for channels IC1 and IC2 are opposite, by setting ATIMER_CCER.CC1P = 0 and ATIMER_CCER.CC2P = 1.
5. Configure the input filter time by setting ATIMER_CCMR1.IC1F[3:0] and ATIMER_CCMR1.IC2F[3:0].
6. Configure the input prescaler by setting ATIMER_CCMR1.IC1PS[1:0] and ATIMER_CCMR1.IC2PS[1:0].
7. Select the trigger input signal by setting ATIMER_SMCR.TS[2:0] = 101.
8. Configure the slave mode controller to reset mode by setting ATIMER_SMCR.SMS[2:0] = 100.
9. Enable the channels by setting ATIMER_CCER.CC1E = 1 and ATIMER_CCER.CC2E = 1.

12.4.15 Software Force Output

In output compare mode, the software can directly force OCxREF to a specific level, independent of the comparison results of ATIMER_CCR and ATIMER_CNT.

The OCxREF signal can be forced to be active (OCxREF is fixed at a high level) by writing OCxM = 101, and forced to be inactive (low level) by writing OCxM = 100. However, the software force operation does not cancel the comparison process; the comparison between ATIMER_CCR and ATIMER_CNT continues.

12.4.16 Output Compare Mode

In output compare mode, when a match is found between ATIMER_CCR and ATIMER_CNT,

OCxREF can be set to active, inactive or toggle on match. At the same time, the interrupt flag is set and a DMA request can be sent (by rewriting the configuration register).

Output compare can also be used to generate a pulse signal of a specific width (one-shot output). The steps are as follows:

1. Select the counting clock (internal, external, prescaled, etc.).
2. Write the desired data to the ATIMER_ARR and ATIMER_CCR registers.
3. Enable interrupts and DMA as required.
4. Select the output mode.
5. Enable the counter.

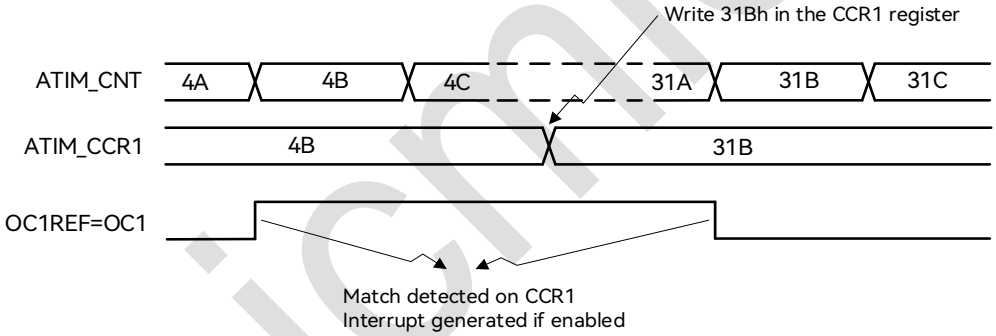


Figure 12-28: Output Compare Mode, Toggle OC1

Without enabling preload, the software can rewrite the ATIMER_CCR register at any time to achieve real-time control of the output waveform. If preload is enabled, the shadow register of ATIMER_CCR will only be updated with the content of the preload register during the next update event.

12.4.17 PWM Output

PWM mode can output pulse width modulation signals with the period determined by the ATIMER_ARR register and the duty cycle determined by the ATIMER_CCR register.

The polarity of the output signal can be configured by the CCxP bit in the register. In PWM mode,

ATIMER_CNT and ATIMER_CCR are compared in real time. Since the counter supports both edge-aligned and center-aligned counting modes, PWM output also supports both edge-aligned and center-aligned modes.

- PWM edge-aligned mode

  In up-counting mode, when it is configured in PWM mode 1, the OCxREF signal is high when ATIMER_CNT < ATIMER_CCR, otherwise it is low. And OCxREF will be fixed at 1 if ATIMER_CCR > ATIMER_ARR, and fixed at 0 if CCR is 0.

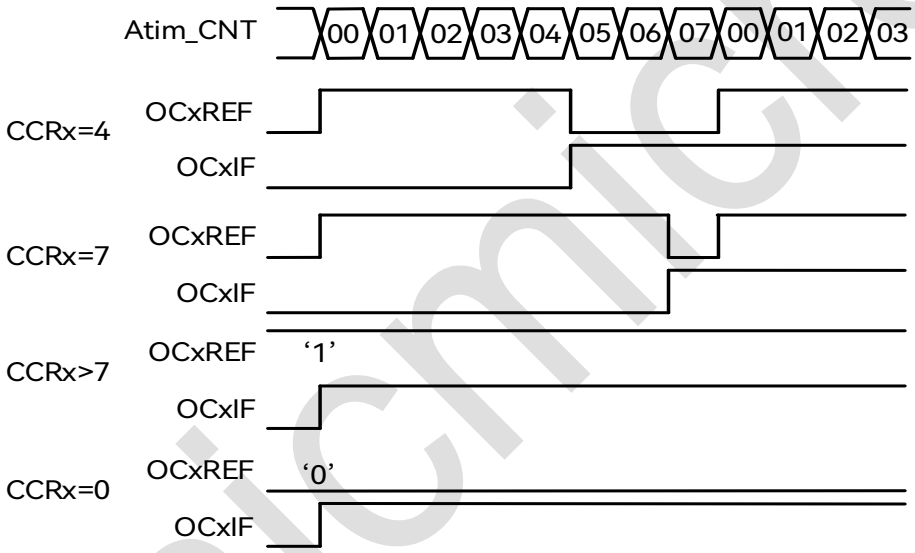


Figure 12-29: Edge-aligned PWM Waveform (ARR = 7)

In down-counting mode, the definition of OCxREF level is the same as that in up-counting mode.

- PWM center-aligned mode

  The definition of OCxREF level is the same as that in edge-aligned mode. The figure below is an example.

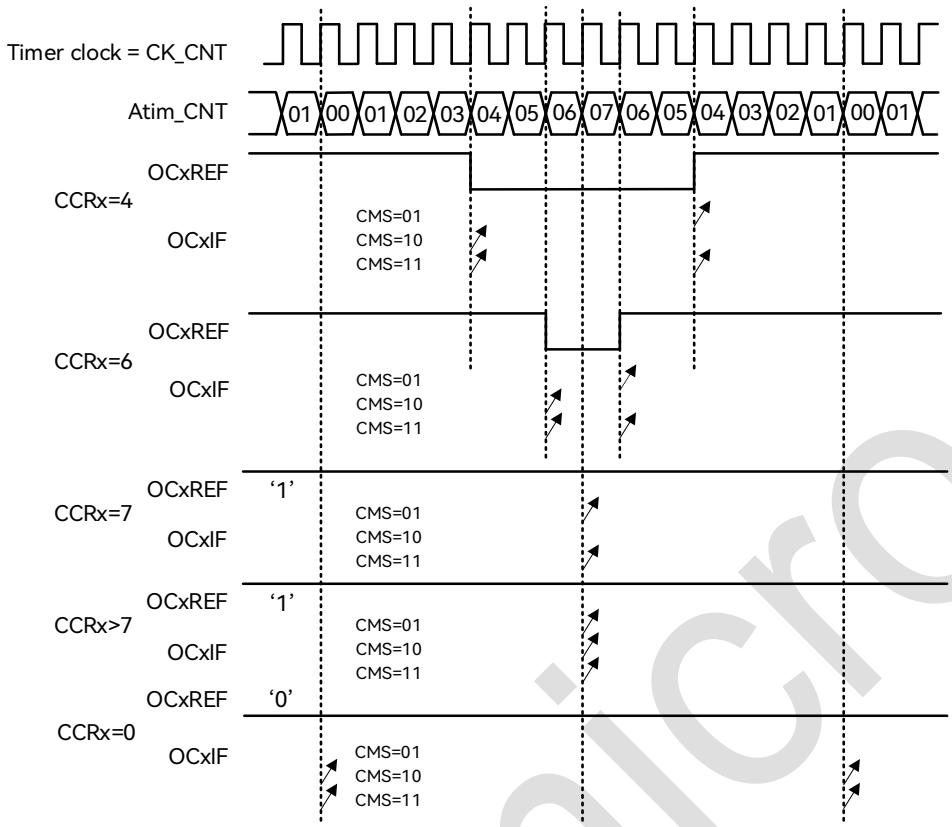


Figure 12-30: Center-aligned PWM Waveform (ARR = 7)

When start counting in center-aligned mode, the initial counting direction is determined by the DIR bit in the register, and in the subsequent process, the DIR bit is directly controlled by hardware. The safest way to use center-aligned mode is to generate an update by setting the UG bit in the register just before starting the counter and not to overwrite the counter while it is running.

12.4.18 Asymmetric PWM Phase-Shift Control

The methods for controlling PWM signals mainly include adjusting the period, duty cycle, and phase. The adjustment strategy shall be tailored to the specific circuit and application scenario. For example, in power conversion applications, the output voltage or current is generally controlled by adjusting the duty cycle. In motor control, in addition to the duty cycle, PWM phase shifting can be used to generate various output waveforms, thereby optimizing motor performance.

In central-aligned mode, ATIMER can generate PWM signals with programmable phase shift. Channels 1–3 support the phase-shift function of PWM. ATIMER_PMEN[2:0] is used to enable the PWM phase-shift output of different channels. The period of the PWM is determined by the ATIMER_ARR register, while the duty cycle is determined by ATIMER_CCR and ATIMER_PMC (compared with ATIMER_CCR during up-counting of ATIMER_CNT and with ATIMER_PMC during down-counting), thereby achieving real-time control of the PWM output.

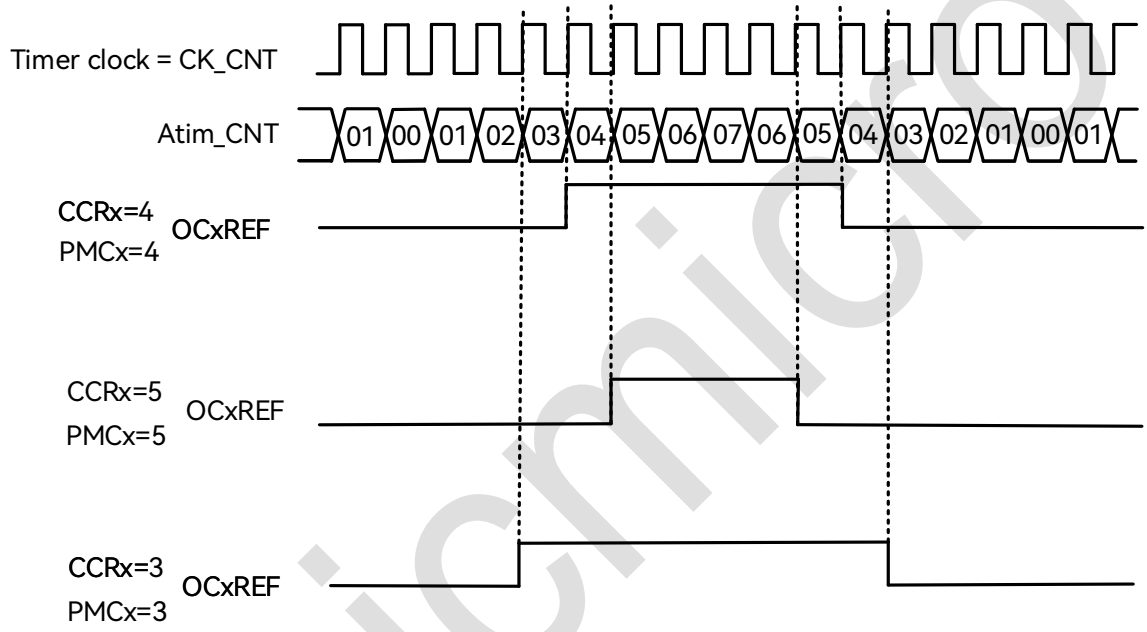


Figure 12-31: Phase-shifted PWM Waveform (ARR = 7)

12.4.19 Complementary Output and Dead-time Insertion

Channels 1–3 of ATIMER support complementary output and dead-time insertion. The DTG[7:0] bits in the register are used to set the dead-time (effective for all channels simultaneously). The output signal OCx is in phase with the reference signal OCxREF, with the rising edge delayed relative to the reference rising edge. The output signal OCxN is in the opposite phase with the reference signal OCxREF, with the rising edge delayed relative to the reference falling edge.

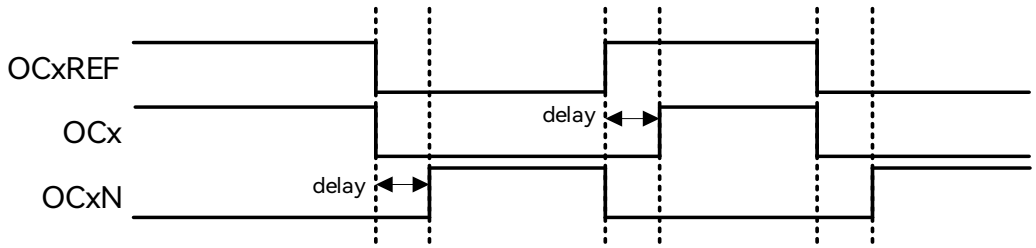


Figure 12-32: Complementary Output with Dead-time Insertion

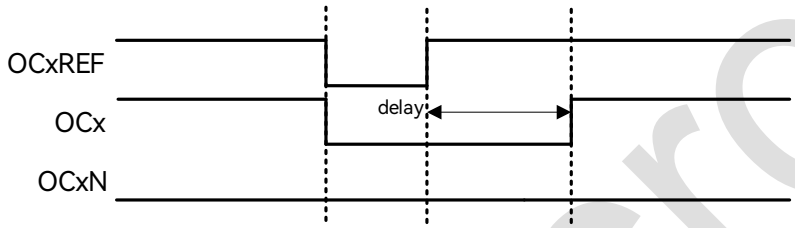


Figure 12-33: Dead-time Waveform with Delay Greater than Negative Pulse

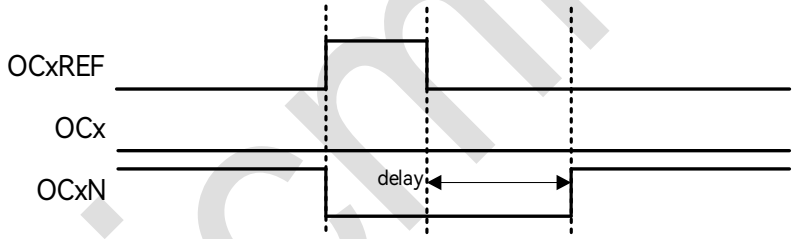


Figure 12-34: Dead-time Waveform with Delay Greater than Positive Pulse

12.4.20 Break Function

The break function can utilize the seven break signals input through the external BRK pin, or effective outputs generated by comparators, SVD, or XTHF oscillator failure detection. After power-on reset, the break circuit is disabled, and the user enables the break function by setting the BKE register. The seven break inputs can be configured for AND or OR operations. The combined break signal can be configured for active polarity and digital filtering.

ATIMER_BRKx is multiplexed with GPIO function: when GPIO is set for digital peripheral function, its input signals is directly connected to the break input of ATIMER; when GPIO is set for other

functions, the break input port of ATIMER is fixed to 1. The actual level of the gated BRKx signal can be controlled by the BRKxGATE register, and the software can flexibly set the unused BRKx to low (0) or high (1) levels to meet the needs of the downstream logic circuit.

When a break event occurs:

- The output enable register is asynchronously cleared, and the output can be forced to inactive, idle or reset state through the OSSI bit.
- Each output channel is driven to the level defined by the OISx bit in the register.
- When complementary output is enabled, the output is asynchronously set to the inactive and reset states, and the dead-time insertion circuit begins to operate, driving the output to the levels defined by OISx and OISxN after the dead-time.
- The break flag register is set, which can trigger an interrupt based on configuration.
- If automatic output is enabled (AOE = 1), the output enable bit (MOE) will be automatically set at the next update event; otherwise, MOE will remain 0 until it is reset by software.

Note: The BRK signal is level-sensitive. Therefore, when BRK remains active, MOE cannot be enabled, and the brake flag BIF cannot be cleared.

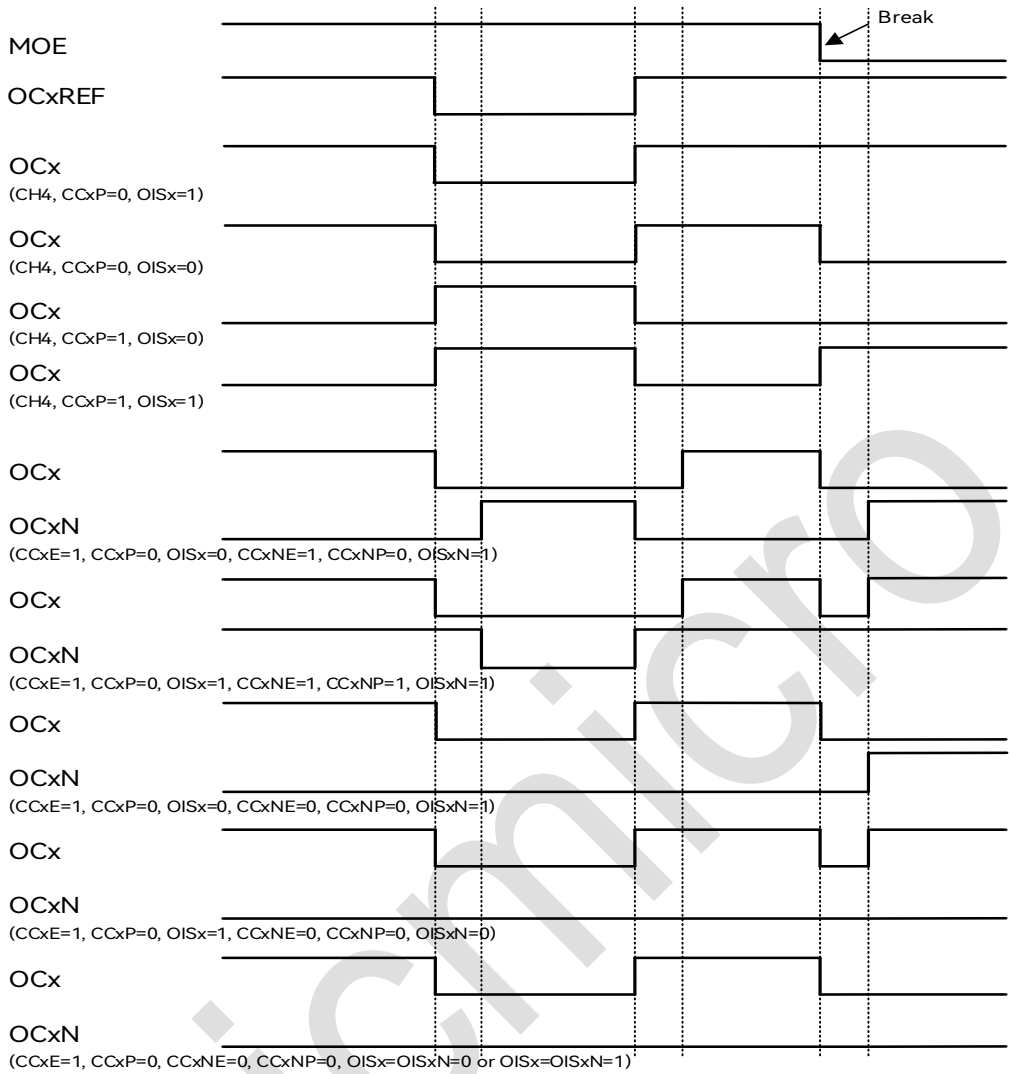


Figure 12-35: Output in Response to Break

12.4.21 6-step PWM Output

For a channel configured in complementary-output mode, the OCxM, CCxE and CCxNE bits are shadowed; their preload values are transferred to the active bits on the COM (commutation) event. Users can thus pre-program the next configuration and have all channels updated synchronously upon a COM event. A COM event can be generated by software (setting the COM bit in ATIMER_EGR) or by a rising edge on TRGI.

When a COM event occurs, the commutation-flag register is set and can generate an interrupt or a DMA request.

The figure below shows an example of six-step commutation; three cases illustrate how the outputs change under different configurations when the COM event occurs.

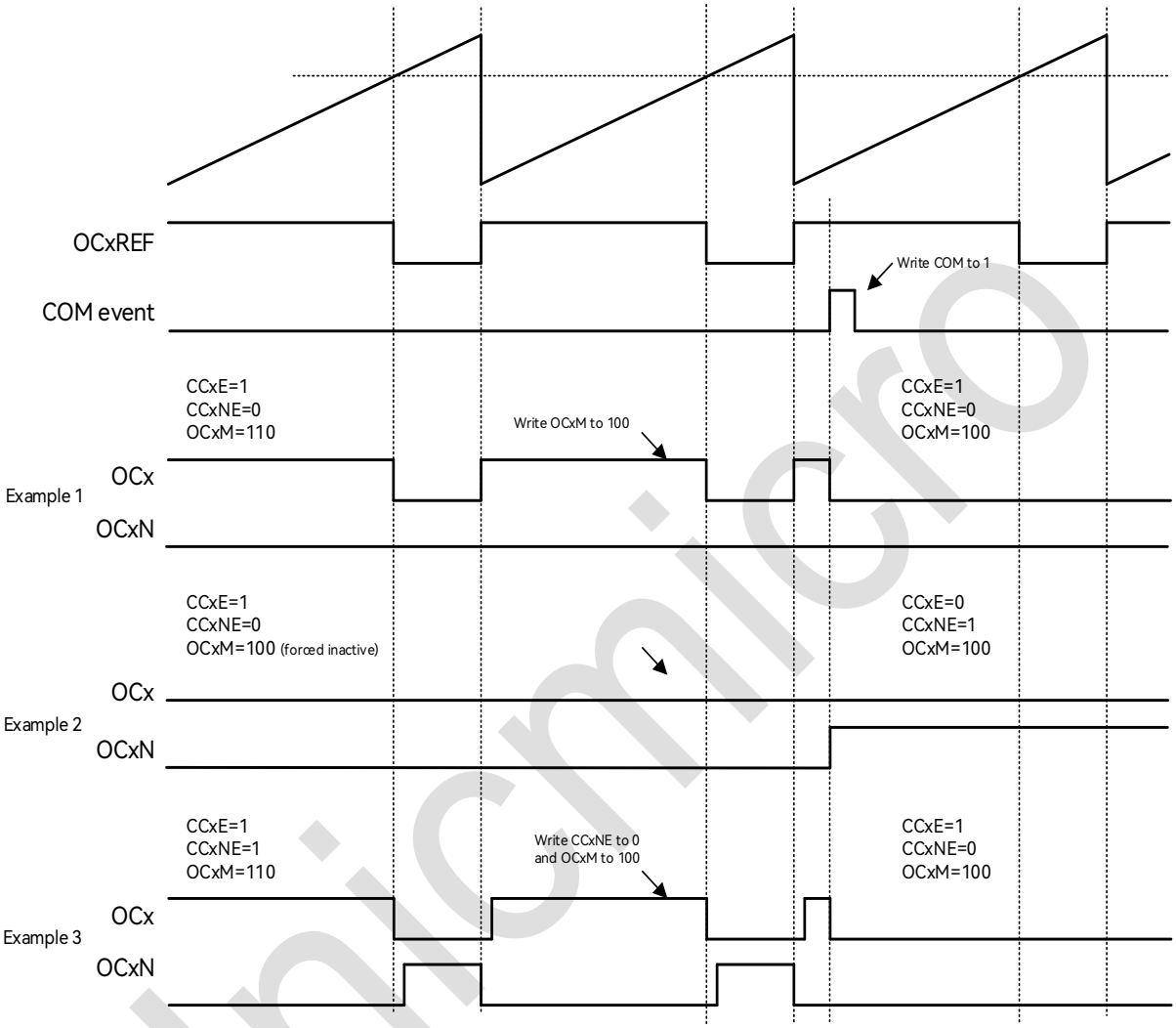


Figure 12-36: Example of Generating Six-step PWM Using COM (OSSR = 1)

12.4.22 Single-pulse Output

Single-pulse output is a special case of the compare output mode, allowing users to output a pulse signal with a programmable width after a programmable delay following the occurrence of a specified event.

Unlike other output modes, the counter will automatically stop at the next update event. A pulse can be correctly generated only if the initial values of ATIMER_CCR and ATIMER_CNT are

different. In up-counting, it is required that ATIMER_CNT < ATIMER_CCR ≤ ATIMER_ARR; while in down-counting, it is required that ATIMER_CNT > ATIMER_CCR.

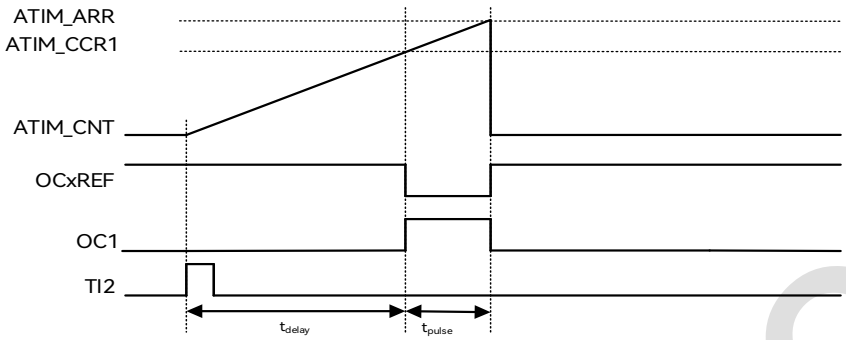


Figure 12-37: Example of Single-pulse Mode

The figure above shows that TI2 input is used as the counter trigger signal. When the counter value equals ATIMER_CCR, OCxREF outputs a low level. When the counter reaches ATIMER_ARR, OCxREF returns to a high level, and the counter rolls back to 0 and stops counting.

The configuration for TI2 serving as the input trigger is as follows:

1. In GPIO module, configure the corresponding pin as ATIMER_CH2.
2. Disable the channel by setting ATIMER_CCER.CC2E = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting ATIMER_CCMR1.CC2S = 01.
4. Select the active counting edge by setting ATIMER_CCER.CC2P = 0.
5. Select TI2FP2 as the trigger input signal (TRGI) by setting ATIMER_SMCR.TS[2:0] = 110.
6. Set the slave mode controller to trigger mode by setting ATIMER_SMCR.SMS[2:0] = 110, with I2FP2 for activating the counter.
7. Enable the channel by setting ATIMER_CCER.CC2E = 1.

The configuration for OC1 serving as the output is as follows:

1. In GPIO module, configure the corresponding pin as ATIMER_CH1.

2. Disable the channel by setting ATIMER_CCER.CC1E = 0 to ensure the success of subsequent channel configuration.
3. Configure the output channel by setting ATIMER_CCMR1.CC1S = 00.
4. Select the active counting edge by setting ATIMER_ CCMR1.OC1M = 111, for PWM mode 2.
5. Enable the channel by setting ATIMER_CCER.CC1E = 1.

Special settings for generating OPM waveform timing:

1. t_{delay} is determined by the value of ATIMER_CCR1.
2. t_{pulse} is determined by the difference between ATIMER_ARR and ATIMER_CCR1 (ATIMER_ARR - ATIMER_CCR1).
3. Configure to single-pulse mode by setting ATIMER_CR1.OPM = 1.

12.4.23 External Event Clearing OCxREF

OCxREF is active at high level, and it can be pulled down directly until the next update event by applying a high level to the external ETR pin. This function is effective only in output compare and PWM modes, and does not work in software force mode. To enable this function, OCxCE must be set to 1.

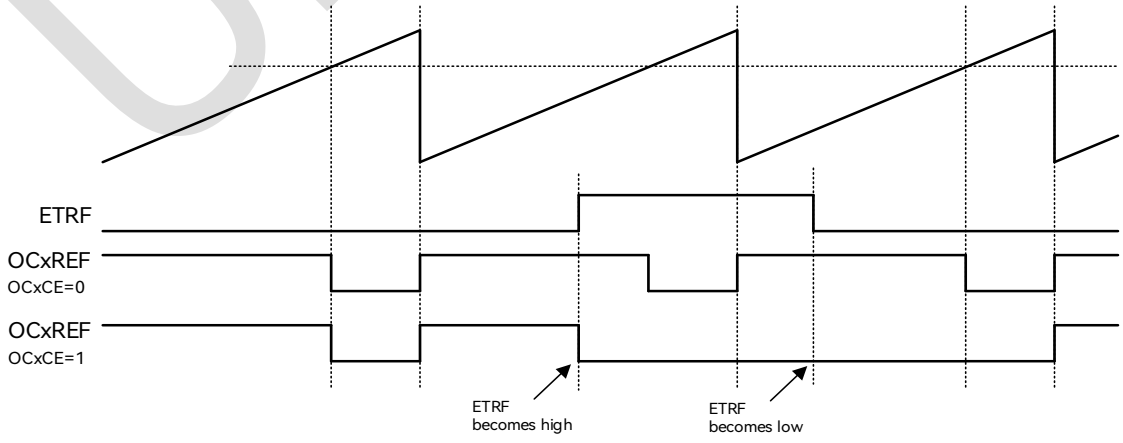


Figure 12-38: ETR Signal Clearing OCxREF of ATIMER

12.4.24 Encoder Interface Mode

The encoder interface mode involves two external input signals. ATIMER determines whether to increment or decrement the counter value based on the edge of one signal relative to the level of the other signal. The following table shows the relationship between the counting modes and the two input signals:

Table 12-2: Counting Direction versus Encoder Signals

Active Edge	Corresponding Signal Level (TI1 corresponds to TI2, and TI2 corresponds to TI1)	TI1 Signal		TI2 Signal	
		Rising	Falling	Rising	Falling
Count only on TI1	High	Decrement	Increment	No count	No count
	Low	Up	Decrement	No count	No count
Counting on TI2 only	High	No count	No count	Up	Down
	Low	No count	No count	Decrement	Increment
Count on both TI1 and TI2	High	Down	Increment	Increment	Down
	Low	Increment	Down	Decrement	Increment

For example, when the counter counts using TI1 as the clock: if TI2 is sampled as high at the rising edge of TI1, the counter will decrement; if TI2 is sampled as high at the falling edge of TI1, the counter will increment.

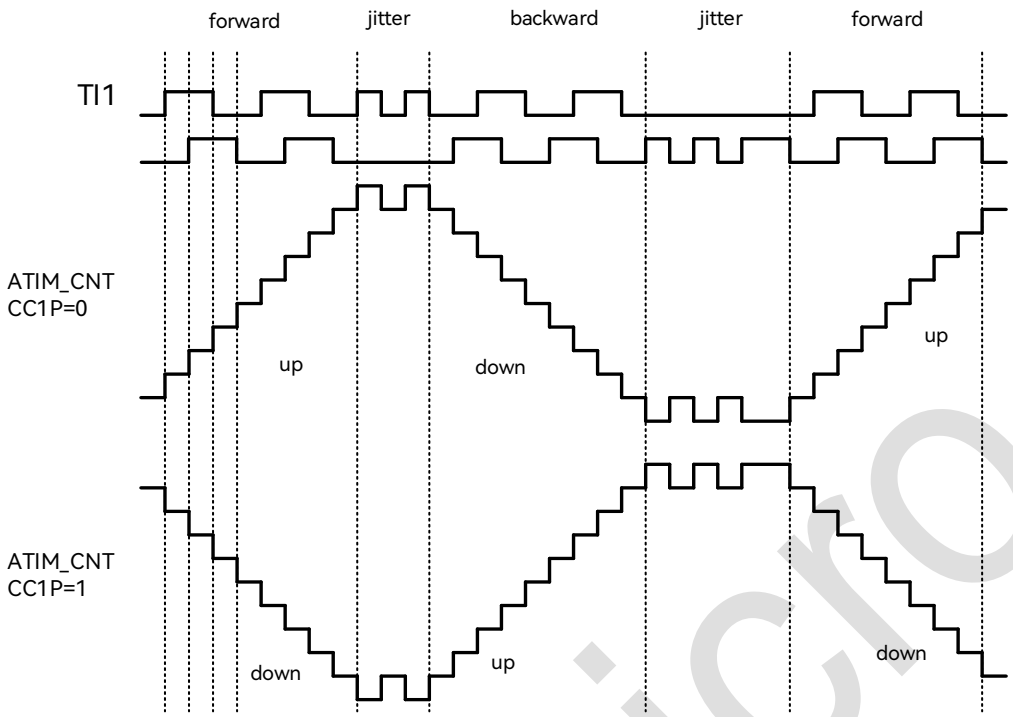


Figure 12-39: Example of Counter Operation in Encoder Mode

The input channels in encoder mode shall be configured as follows:

1. In GPIO module, configure the corresponding pins with ATIMER_CH1 and ATIMER_CH2 functions.
2. Disable the channel by setting ATIMER_CCER.CC1E = 0 and ATIMER_CCER.CC2E = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting ATIMER_CCMR1.CC1S = 01 and ATIMER_CCMR1.CC2S = 01.
4. Select the active counting edge by setting ATIMER_CCER.CC1P = 0 and ATIMER_CCER.CC2P = 0.
5. Set the slave mode controller to encoder mode 3 by setting ATIM_SMCR.SMS[2:0] = 011.
6. Enable the channels by setting ATIMER_CCER.CC1E = 1 and ATIMER_CCER.CC2E = 1.

12.4.25 ATIMER Slave Mode

When ATIMER operates as a slave (triggered by external events), it can be configured in three working modes: reset mode, gated mode, and trigger mode.

12.4.26 Reset Mode

In this mode, all the preload registers inside ATIMER will be reinitialized due to external input events, and the CNT will return to 0 and start counting again. The following figure shows that the counter counts normally until the rising edge of external TI1 input occurs, at which time the counter is cleared and then restarts counting. The configuration steps are as follows:

1. In GPIO module, configure the corresponding pin as ATIMER_CH1.
2. Disable the channel by setting ATIMER_CCER.CC1E = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting ATIMER_CCMR1.CC1S = 01.
4. Select the active counting edge by setting ATIMER_CCER.CC1P = 0.
5. Select TI1FP1 as the trigger input signal (TRGI) by setting ATIMER_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to reset mode by setting ATIMER_SMCR.SMS[2:0] = 100.
7. Enable the channel by setting ATIMER_CCER.CC1E = 1.
8. Enable the counter by setting ATIMER_CR1.CEN = 1.

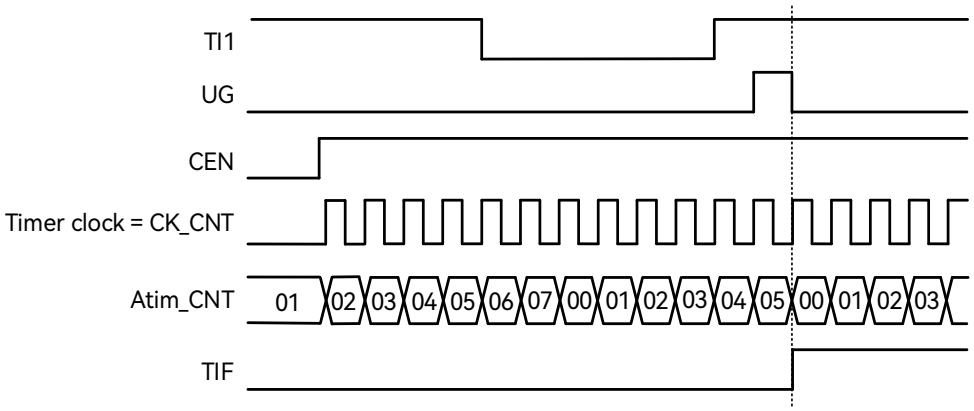


Figure 12-40: Timing Diagram in Reset Mode

12.4.27 Gated Mode

In gated mode, the counter operates only when the input signal is at a specific level. Level transitions that cause the counter to start or stop counting will trigger an interrupt flag. The configuration steps in the example are as follows:

1. In GPIO module, configure the corresponding pin as ATIMER_CH1.
2. Disable the channel by setting ATIMER_CCER.CC1E = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting ATIMER_CCMR1.CC1S = 01.
4. Select the active counting edge by setting ATIMER_CCER.CC1P = 1.
5. Select TI1FP1 as the trigger input signal (TRGI) by setting ATIMER_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to gated mode by setting ATIMER_SMCR.SMS[2:0] = 101.
7. Enable the channel by setting ATIMER_CCER.CC1E = 1.
8. Enable the counter by setting ATIMER_CR1.CEN = 1.

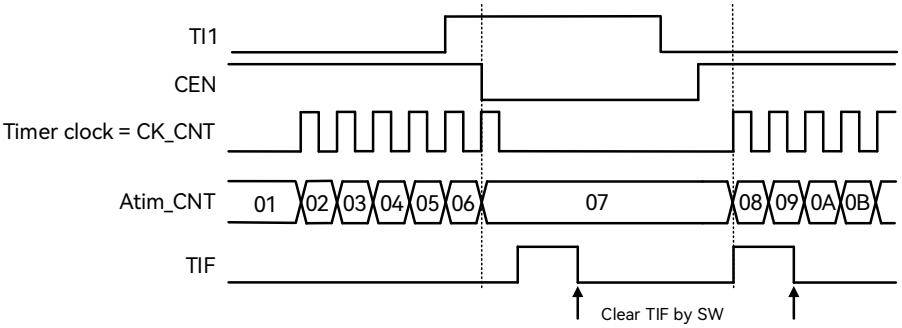


Figure 12-41: Timing Diagram in Gated Mode

12.4.28 Trigger Mode

The counter starts counting only after a specific external input event occurs. The configuration steps in the example are as follows:

1. In GPIO module, configure the corresponding pin as ATIMER_CH1.
2. Disable the channel by setting ATIMER_CCER.CC1E = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting ATIMER_CCMR1.CC1S = 01.
4. Select the active counting edge by setting ATIMER_CCER.CC1P = 0.
5. Select TI1FP1 as the trigger input signal (TRGI) by setting ATIMER_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to trigger mode by setting ATIMER_SMCR.SMS[2:0] = 110.
7. Enable the channel by setting ATIMER_CCER.CC1E = 1.

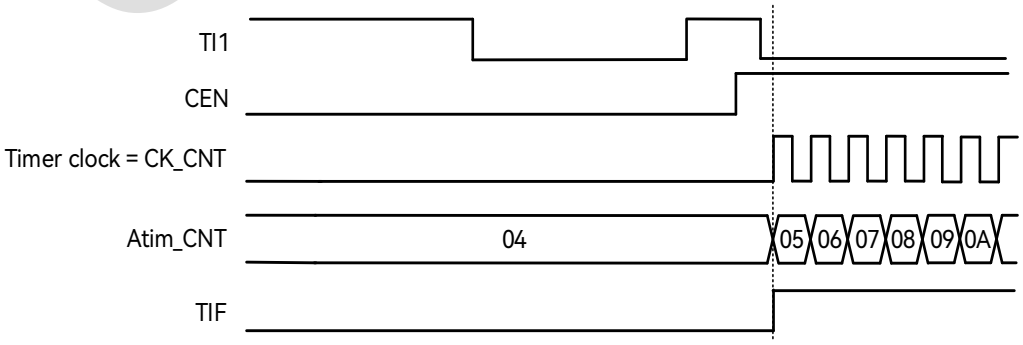


Figure 12-42: Timing Diagram in Trigger Mode

12.4.29 External Clock Counting Mode Triggered by External Events

In this mode, ETR can be set as the counting clock, while another external input is used as the trigger signal to start the counter. For instance, the counter begins counting on the rising edge of the ETR input after detecting the rising edge of TI1. The configuration steps are as follows:

1. In GPIO module, configure the corresponding pins for ATIMER_CH1 and ATIMER_ETR functions.
2. Configure ETP for edge selection by setting ATIMER_SMCR.ETP = 0.
3. Configure the ETR prescaler ratio by setting ATIMER_SMCR.ETPS[1:0] = 01.
4. Configure the input filter time by setting ATIMER_SMCR.ETF[3:0] = 0000.
5. Set the ECE register and enable the external clock mode 2 by setting ATIMER_SMCR.ECE = 1.
6. Disable the channel by setting ATIMER_CCER.CC1E = 0 to ensure the success of subsequent channel configuration.
7. Select the input channel by setting ATIMER_CCMR1.CC1S = 01.
8. Select the active counting edge by setting ATIMER_CCER.CC1P = 0.
9. Select TI1FP1 as the trigger input signal (TRGI) by setting ATIMER_SMCR.TS[2:0] = 101.
10. Configure the slave mode controller to trigger mode by setting ATIMER_SMCR.SMS[2:0] = 110.
11. Enable the channel by setting ATIMER_CCER.CC1E = 1.

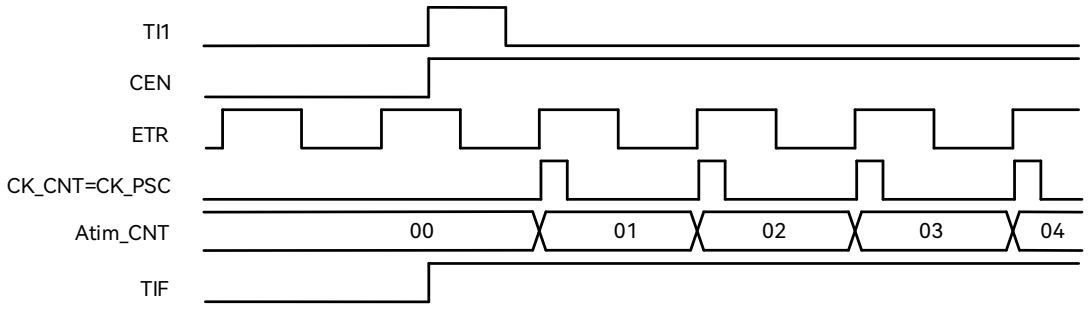


Figure 12-43: Timing Diagram in External Clock Mode 2 and Trigger Mode

12.4.30 DMA Access

ATIMER supports seven types of DMA requests, namely four CC channel requests, external trigger request, update event trigger request and COM trigger request.

Each CC channel generates a DMA request, which is used to transfer the content of ATIMER_CCRx to RAM in capture mode, and to write the data from RAM to ATIMER_CCRx in compare mode. The DMA requests for CC channels can be configured for single transfers or burst transfers (CCxBURSTEN), wherein the former can access only the ATIMER_CCRx register, while the latter can access a specific set of registers based on configurations of the ATIMER_DCR register.

In addition, DMA requests can also be generated from external trigger events, update events and COM events. At the occurrence of these requests, DMA burst transfer will be initiated to write data to one or more registers inside ATIMER or to read one or more register values from ATIMER.

Table 12-3: DMA Access Counting Mode

DMA Request	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA Access Object	Single-transfer Length
ATIMER_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_CH2	0	0	Read CCR2	1
		1	Write CCR2	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_CH4	0	0	Read CCR4	1

DMA Request	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA Access Object	Single-transfer Length
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_COM	N/A	0	Read DMAR	DBL
		1	Write DMAR	

12.4.31 Input XOR Function

The input signals of channels 1–3 can be XORed together and then connected to the input of the filtering and edge detection circuit for input capture or triggering of channel 1.

The TI1S bit in the ATIMER_CR2 register is used to select whether the input of channel 1 comes from the XOR of the three channel inputs.

12.4.32 Debug Mode

When entering debug mode, the timer can be stopped or continued, with its behavior defined by the STOP_EN bit.

When the timer is stopped in debug mode, its output will be disabled (MOE set to 0). Depending on the register configuration, the output signal at this time can be forced to inactive or controlled by the GPIO module.

12.4.33 DMA Burst

ATIMER supports DMA and DMA-burst access, allowing for the configuration of DMA requests to be triggered by specific events. This enables the capture results from CCR to be written to

RAM, or the contents of one or more registers from RAM to be written to the preload registers of ATIMER.

DMA-burst allows for multiple successive DMA requests triggered by a single event, primarily for the purpose of continuously updating the contents of multiple registers after an event occurs, thus facilitating dynamic real-time adjustments to output waveforms and other functions.

The DMA controller must point the peripheral target address to a virtual register ATIMER_DMAR. When a specific timer event occurs, ATIMER will continuously issue multiple DMA requests. Each DMA write operation to ATIMER_DMAR will be redirected by ATIMER to the actual functional register.

The DBL register is used to set the DMA burst length, and the DBA register is used to set the base address (relative to the offset of ATIMER_CR) for DMA access to ATIMER.

In DMA-burst mode, all DMA accesses shall be directed to the DMAR virtual register, with the ATIMER automatically accumulating the internal offset address based on the accesses. The DBA register specifies the target address for the first DMA transfer within ATIMER, while the DBL register specifies the burst length.

12.5 Register Description

ATIMER register base address: 0x40007400

Table 12-4: List of ATIMER Registers

Offset	Name	Description
0x00	ATIMER_CR1	ATIMER control register 1
0x04	ATIMER_CR2	ATIMER control register 2
0x08	ATIMER_SMCR	ATIMER slave mode control register
0x0C	ATIMER_DIER	ATIMER DMA and interrupt enable register
0x10	ATIMER_SR	ATIMER status register
0x14	ATIMER_EGR	ATIMER event generation register
0x18	ATIMER_CCMR1	ATIMER capture/compare mode register 1

Offset	Name	Description
0x1C	ATIMER_CCMR2	ATIMER capture/compare mode register 2
0x20	ATIMER_CCER	ATIMER capture/compare enable register
0x24	ATIMER_CNT	ATIMER counter register
0x28	ATIMER_PSC	ATIMER prescaler register
0x2C	ATIMER_ARR	ATIMER auto-reload register
0x30	ATIMER_RCR	ATIMER repetition counter register
0x34	ATIMER_CCR1	ATIMER capture/compare register 1
0x38	ATIMER_CCR2	ATIMER capture/compare register 2
0x3C	ATIMER_CCR3	ATIMER capture/compare register 3
0x40	ATIMER_CCR4	ATIMER capture/compare register 4
0x44	ATIMER_BDTR	ATIMER break and dead-time control register
0x48	ATIMER_DCR	ATIMER DMA control register
0x4C	ATIMER_DMAR	ATIMER DMA access register
0x50	ATIMER_CCMR3	ATIMER compare mode register 3
0x54	ATIMER_CCR5	ATIMER compare register 5
0x58	ATIMER_PMEN	ATIMER PWM phase-shift enable register
0x5C	ATIMER_PMC1	ATIMER phase-shift offset register 1
0x60	ATIMER_PMC2	ATIMER phase-shift offset register 2
0x64	ATIMER_PMC3	ATIMER phase-shift offset register 3
0x68	ATIMER_BKCTL	ATIMER break input control register

12.5.1 ATIMER Control Register 1 ATIMER_CR1 (Offset: 00H)

Bit	Name	Attribute	Reset Value	Description
31:18	RSV	-	-	Reserved
17	STOP_EN	RW	0x1	ATIMER break enable bit: after software sets a breakpoint, the counter is prohibited from counting: 0: Counter operates normally. 1: Counter stops counting.
16	DBOEN	RW	0x0	Three-channel complementary output enable control register in debug mode: 0: In debug mode, the three-channel complementary outputs are in high-impedance state. 1: In debug mode, the level of the three-channel complementary outputs is determined by the DBCC*

Bit	Name	Attribute	Reset Value	Description
				registers.
15	DBCC3N	RW	0x0	OC3N output register in debug mode: 0: Output 0 in debug mode. 1: Output 1 in debug mode.
14	DBCC3	RW	0x0	OC3 output register in debug mode: 0: Output 0 in debug mode. 1: Output 1 in debug mode.
13	DBCC2N	RW	0x0	OC2N output register in debug mode: 0: Output 0 in debug mode. 1: Output 1 in debug mode.
12	DBCC2	RW	0x0	OC2 output register in debug mode: 0: Output 0 in debug mode. 1: Output 1 in debug mode.
11	DBCC1N	RW	0x0	OC1N output register in debug mode: 0: Output 0 in debug mode. 1: Output 1 in debug mode.
10	DBCC1	RW	0x0	OC1 output register in debug mode: 0: Output 0 in debug mode. 1: Output 1 in debug mode.
9:8	CKD	RW	0x0	Dead time and digital filter clock frequency division register (division ratio relative to CK_INT): 00: $t_{DTS} = t_{CK\_INT}$ 01: $t_{DTS} = 2 * t_{CK\_INT}$ 10: $t_{DTS} = 4 * t_{CK\_INT}$ 11: Reserved for future use (RFU), disabled
7	ARPE	RW	0x0	Auto-reload preload enable: 0: ARR register does not enable preload. 1: ARR register enables preload.
6:5	CMS	RW	0x0	Counter alignment mode selection: 00: Edge-aligned mode 01: Center-aligned mode 1, output compare interrupt flag is set only during down-counting 10: Center-aligned mode 2, output compare interrupt flag is set only during up-counting 11: Center-aligned mode 3, output compare

Bit	Name	Attribute	Reset Value	Description
				interrupt flag is set during both up-counting and down-counting
4	DIR	RW	0x0	Counting direction register: 0: Up-counting 1: Down-counting Note: This register is read-only when the timer is configured in center-aligned mode or encoder mode.
3	OPM	RW	0x0	One-pulse output mode: 0: The counter does not stop at the occurrence of an update event. 1: The counter stops (CEN automatically cleared) at the occurrence of an update event.
2	URS	RW	0x0	Update request selection: 0: An update interrupt or DMA request can be generated by any of the following events: - Counter overflow/underflow - Software setting the UG bit - Slave controller generating an update 1: An update interrupt or DMA request can be generated only at counter overflow or underflow.
1	UDIS	RW	0x0	Update disable: 0: Enable update event; an update event can be generated by any of the following events: - Counter overflow/underflow - Software setting the UG bit - Slave controller generating an update 1: Disable update event, do not update shadow register. The counter and prescaler will be reinitialized if the UG bit is set or the slave controller receives a hardware reset.
0	CEN	RW	0x0	Counter enable: 0: Counter is disabled. 1: Counter is enabled. Note: External trigger mode can automatically set CEN.

12.5.2 ATIMER Control Register 2 ATIMER_CR2 (Offset: 04H)

Bit	Name	Attribute	Reset Value	Description
31:19	RSV	-	-	Reserved
18	CC4BUEN	RW	0x0	DMA mode configuration for capture/compare channel 4: 0: Single transfer, DMA reads the value of the CCR4 register and transfers it to RAM. 1: Burst mode from RAM to ATIMER, with access address and length configured via DCR.
17	CC3BUEN	RW	0x0	DMA mode configuration for capture/compare channel 3: 0: CCR3 to RAM, DMA reads the value of the CCR3 register and transfers it to RAM. 1: Burst mode from RAM to ATIMER, with access address and length configured via DCR.
16	CC2BUEN	RW	0x0	DMA mode configuration for capture/compare channel 2: 0: CCR2 to RAM, DMA reads the value of the CCR2 register and transfers it to RAM. 1: Burst mode from RAM to ATIMER, with access address and length configured via DCR.
15	CC1BUEN	RW	0x0	DMA mode configuration for capture/compare channel 1: 0: CCR1 to RAM, DMA reads the value of the CCR1 register and transfers it to RAM. 1: Burst mode from RAM to ATIMER, with access address and length configured via DCR.
14	OIS4	RW	0x0	Referring to OIS1.
13	OIS3N	RW	0x0	Referring to OIS1N.
12	OIS3	RW	0x0	Referring to OIS1.
11	OIS2N	RW	0x0	Referring to OIS1N.
10	OIS2	RW	0x0	Referring to OIS1.
9	OIS1N	RW	0x0	Definition of OC1N output idle state: 0: When MOE = 0, OC1N = 0 after the dead time. 1: When MOE = 0, OC1N = 1 after the dead time.

Bit	Name	Attribute	Reset Value	Description
8	OIS1	RW	0x0	Definition of OC1 output idle state: 0: When MOE = 0 (if complementary output is enabled, after the dead time), OC1 = 0. 1: When MOE = 0 (if complementary output is enabled, after the dead time), OC1 = 1.
7	TI1S	RW	0x0	ATIMER input TI1 selection: 0: The ATIMER_CH1 pin is connected to TI1 input. 1: The ATIMER_CH1, CH2 and CH3 pins are connected to TI1 input via XOR.
6:4	MMS	RW	0x0	Master mode selection for configuring the source of the synchronization trigger signal (TRGO) sent to slave in master mode: 000: The UG bit of ATIMER_EGR is used as TRGO. 001: The counter enable signal CNE_EN is used as TRGO, which can be used to start multiple timers simultaneously. 010: The UE (update event) signal is used as TRGO. 011: Compare pulse, if the CC1IF flag is about to set, TRGO outputs a positive pulse. 100: OC1REF is used as TRGO. 101: OC2REF is used as TRGO. 110: OC3REF is used as TRGO. 111: OC4REF is used as TRGO. Note: The slave timer or ADC must have its working clock enabled in advance to receive the TRGO sent by the master timer.
3	CCDS	RW	0x0	Capture/compare DMA selection: 0: Send DMA request when a capture/compare event occurs. 1: Send DMA request when an update event occurs.
2	CCUS	RW	0x0	Capture/compare control register update selection: 0: When the capture/compare control register enables preload (CCPC = 1), it is updated only when the COMG register is set. 1: When the capture/compare control register enables preload (CCPC = 1), it is updated either

Bit	Name	Attribute	Reset Value	Description
				when the COMG register is set or on the rising edge of TRGI.
1	RSV	-	-	Reserved
0	CCPC	RW	0x0	Capture/compare preload control: 0: CCxE, CCxNE and OCxM do not enable preload. 1: CCxE, CCxNE and OCxM enable preload. Note: This register is only effective on channels with complementary output.

12.5.3 ATIMER Slave Mode Control Register ATIMER_SMCR (Offset: 08H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	ETP	RW	0x0	External trigger signal polarity configuration: 0: Active on high level or rising edge 1: Active on low level or falling edge
14	ECE	RW	0x0	External clock enable: 0: Disable external clock mode 2 1: Enable external clock mode 2, with the counter clocked by any the active edge of ETRF
13:12	ETPS	RW	0x0	External trigger signal prescaler register: The frequency of external trigger signal ETRP can be at most 1/4 of the ATIMER working clock frequency. When the input signal frequency is high, prescaling can be used. 00: No prescaler 01: Divided by 2 10: Divided by 4 11: Divided by 8
11:8	ETF	RW	0x0	External trigger signal filter frequency and length selection: 0000: No filter

Bit	Name	Attribute	Reset Value	Description
				0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS}$ / 2, N = 6 0101: $f_{SAMPLING} = f_{DTS}$ / 2, N = 8 0110: $f_{SAMPLING} = f_{DTS}$ / 4, N = 6 0111: $f_{SAMPLING} = f_{DTS}$ / 4, N = 8 1000: $f_{SAMPLING} = f_{DTS}$ / 8, N = 6 1001: $f_{SAMPLING} = f_{DTS}$ / 8, N = 8 1010: $f_{SAMPLING} = f_{DTS}$ / 16, N = 5 1011: $f_{SAMPLING} = f_{DTS}$ / 16, N = 6 1100: $f_{SAMPLING} = f_{DTS}$ / 16, N = 8 1101: $f_{SAMPLING} = f_{DTS}$ / 32, N = 5 1111: $f_{SAMPLING} = f_{DTS}$ / 32, N = 6 1111: $f_{SAMPLING} = f_{DTS}$ / 32, N = 8
7	MSM	RW	0x0	Master/slave mode selection: 0: No action, to perfectly synchronize the master timer and slave timer via TRGO (when perfectly synchronized, the master register CR2.MMS must be set to 010). 1: Delay the master TRGI trigger signal. Note: This is only applicable to the trigger mode in slave mode (SMS = 110) and is configured by the slave mode.
6:4	TS	RW	0x0	Trigger selection, selecting the trigger source for synchronizing the counter: 000: Internal trigger signal 0 (ITR0) 001: Internal trigger signal 0 (ITR1) 010: Internal trigger signal 0 (ITR2) 011: Internal trigger signal 0 (ITR3) 100: TI1 edge detection (TI1F_ED) 101: Filtered TI1 (TI1FP1) 110: Filtered TI2 (TI2FP2) 111: External trigger input (ETRF) Note: The TS register can only be rewritten when the slave mode is disabled (i.e. SMS = 000).
3	RSV	-	-	Reserved
2:0	SMS	RW	0x0	Slave mode selection:

Bit	Name	Attribute	Reset Value	Description
				000: Slave mode disabled: after CEN is enabled, the prescaler is clocked directly by the internal clock. 001: Encoder mode 1: the counter counts up/down on the TI2FP2 edge depending on the TI1FP1 level. 010: Encoder mode 2: the counter counts up/down on the TI1FP1 edge depending on the TI2FP2 level. 011: Encoder mode 3: the counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the levels of other input signals. 100: Reset mode: the rising edge of TRGI initializes the counter and triggers a register update. 101: Gated mode: the counting clock is enabled when TRGI is high, and stopped when TRGI is low. 110: Trigger mode: the counter starts counting at the rising edge of TRGI (the counter does not reset). 111: External clock mode 1: the rising edge of TRGI directly drives the counter.

12.5.4 ATIMER DMA and Interrupt Enable Register ATIMER_DIER (Offset: 0CH)

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
16	CC5IE	RW	0x0	Capture/compare channel 5 interrupt enable: 0: Disable capture/compare channel 5 interrupt 0: Disable capture/compare channel 5 interrupt
15	RSV	-	-	Reserved
14	TDE	RW	0x0	External trigger DMA request enable: 0: In slave mode, disable DMA requests generated by external trigger events. 1: In slave mode, enable DMA requests generated by external trigger events (can be used to automatically update the preload register).

Bit	Name	Attribute	Reset Value	Description
13	COMDE	RW	0x0	COM event DMA request enable: 0: Disable DMA request generation on COM event occurrence. 1: Enable DMA request generation on COM event occurrence.
12	CC4DE	RW	0x0	Capture/compare channel 4 DMA request enable: 0: Disable CC4 DMA request. 1: Enable CC4 DMA request.
11	CC3DE	RW	0x0	Capture/compare channel 3 DMA request enable: 0: Disable CC3 DMA request 1: Enable CC3 DMA request.
10	CC2DE	RW	0x0	Capture/compare channel 2 DMA request enable: 0: Disable CC2 DMA request. 1: Enable CC2 DMA request.
9	CC1DE	RW	0x0	Capture/compare channel 1 DMA request enable: 0: Disable CC1 DMA request. 1: Enable CC1 DMA request.
8	UDE	RW	0x0	Update event DMA request enable: 0: Disable DMA request generation on update event occurrence. 1: Enable DMA request generation on update event occurrence.
7	BIE	RW	0x0	Break event interrupt enable: 0: Disable break event interrupt. 1: Enable break event interrupt.
6	TIE	RW	0x0	Trigger event interrupt enable: 0: Disable trigger event interrupt. 1: Enable trigger event interrupt.
5	COMIE	RW	0x0	COM event interrupt enable: 0: Disable COM event interrupt. 1: Enable COM event interrupt.
4	CC4IE	RW	0x0	Capture/compare channel 4 interrupt enable: 0: Disable capture/compare channel 4 interrupt. 0: Disable capture/compare channel 4 interrupt.

Bit	Name	Attribute	Reset Value	Description
3	CC3IE	RW	0x0	Capture/compare channel 3 interrupt enable: 0: Disable capture/compare channel 3 interrupt 0: Disable capture/compare channel 3 interrupt.
2	CC2IE	RW	0x0	Capture/compare channel 2 interrupt enable: 0: Disable capture/compare channel 2 interrupt. 0: Disable capture/compare channel 2 interrupt.
1	CC1IE	RW	0x0	Capture/compare channel 1 interrupt enable: 0: Disable capture/compare channel 1 interrupt. 0: Disable capture/compare channel 1 interrupt.
0	UIE	RW	0x0	Update event interrupt enable: 0: Disable update event interrupt. 1: Enable update event interrupt.

12.5.5 ATIMER Status Register ATIMER_SR (Offset: 10H)

Bit	Name	Attribute	Reset Value	Description
31:18	RSV	-	-	Reserved
17	HARDFAULT	W1C	0x0	Hardfault status bit, set by hardware, cleared by software writing 1.
16	CC5IF	W1C	0x0	Capture/compare channel 5 interrupt: Refer to CC1IF.
15:13	RSV	-	-	Reserved
12	CC4OF	W1C	0x0	Capture/compare channel 4 overcapture status: Refer to CC1OF.
11	CC3OF	W1C	0x0	Capture/compare channel 3 overcapture status: Refer to CC1OF.
10	CC2OF	W1C	0x0	Capture/compare channel 2 overcapture status: Refer to CC1OF.
9	CC1OF	W1C	0x0	Capture/compare channel 1 overcapture status: This register is only valid when the corresponding channel is configured in input capture mode. This flag bit is set by hardware and cleared by software writing 1. 0: No overcapture event

Bit	Name	Attribute	Reset Value	Description
				1: A new capture occurs while the CC1IF flag is 1.
8	RSV	-	-	Reserved
7	BIF	W1C	0x0	Break event interrupt flag, set by hardware and cleared by software writing 1.
6	TIF	W1C	0x0	Trigger event interrupt flag, set by hardware and cleared by software writing 1.
5	COMIF	W1C	0x0	COM event interrupt flag, set by hardware and cleared by software writing 1.
4	CC4IF	W1C	0x0	Capture/compare channel 4 interrupt flag: Refer to CC1IF.
3	CC3IF	W1C	0x0	Capture/compare channel 3 interrupt flag: Refer to CC3IF.
2	CC2IF	W1C	0x0	Capture/compare channel 2 interrupt flag: Refer to CC2IF.
1	CC1IF	W1C	0x0	Capture/compare channel 1 interrupt flag: If CC1 channel is configured for output: CC1IF is set when the counter value equals the compare value, and cleared by software writing 1. If CC1 channel is configured for input: CC1IF is set when a capture event occurs, and cleared by software writing 1, or automatically cleared by software reading ATIMER_CCR1.
0	UIF	W1C	0x0	Update event interrupt flag, set by hardware and cleared by software writing 1. UIF is set and the shadow register is updated at the following events: Counter overflow occurs when repetition counter = 0 and UDIS = 0. The counter is initialized by software setting the UG bit when URS = 0 and UDIS = 0. The counter is initialized by a trigger event when URS = 0 and UDIS = 0.

12.5.6 ATIMER Event Generation Register ATIMER_EGR (Offset: 14H)

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
16	CC5G	W	0x0	Capture/compare channel 5 software trigger, refer to CC1G
15:8	RSV	-	-	Reserved
7	BG	W	0x0	Software break event: This bit is set by software to generate a break event, and automatically cleared by hardware.
6	TG	W	0x0	Software trigger event: This bit is set by software to generate a trigger event, and automatically cleared by hardware.
5	COMG	W	0x0	Software COM event: This bit is set by software writing 1 and automatically cleared by hardware.
4	CC4G	W	0x0	Capture/compare channel 4 software trigger, refer to CC1G
3	CC3G	W	0x0	Capture/compare channel 3 software trigger, refer to CC1G
2	CC2G	W	0x0	Capture/compare channel 2 software trigger, refer to CC1G
1	CC1G	W	0x0	Capture/compare channel 1 software trigger, automatically cleared by hardware If CC1 channel is configured for output: CC1IF is set, and if enabled, it can generate the corresponding interrupt and DMA request. If CC1 channel is configured for input: the current counter value is captured into the ATIMER_CCR1 register, CC1IF is set, and if enabled, it can generate the corresponding interrupt and DMA request.
0	UG	W	0x0	Software update event: This bit can be set by software to generate an update event, and is automatically cleared by hardware. When the software sets UG, the counter is reinitialized, the shadow register is updated, and the prescaler counter is cleared.

12.5.7 ATIMER Capture/Compare Mode Register 1 ATIMER_CCMR1 (Offset: 18H)

This register is multiplexed for two different functions under output compare and input capture modes:

1. Output compare mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	OC2CE	RW	0x0	Output compare 2 clear enable, refer to OC1CE
14:12	OC2M	RW	0x0	Output compare 2 mode configuration, refer to OC1M
11	OC2PE	RW	0x0	Output compare 2 preload enable, refer to OC1PE
10	OC2FE	RW	0x0	Output compare 2 fast enable, refer to OC1FE
9:8	CC2S	RW	0x0	Capture/compare channel 2 selection: 00: CC2 channel configured as output. 01: CC2 channel configured as input, IC2 mapped to TI2. 10: CC2 channel configured as input, IC2 mapped to TI1. 11: CC2 channel configured as input, IC2 mapped to TRC. Note: CC2S can only be written when the channel is disabled (CC2E = 0).
7	OC1CE	RW	0x0	Output compare 1 clear enable: 0: OC1REF is not affected by ETRF. 1: OC1REF is automatically cleared when ETRF is high.
6:4	OC1M	RW	0x0	Output compare 1 mode configuration: these bits define the behavior of the OC1REF signal. 000: The comparison result between the output compare register CCR1 and the counter CNT does not affect the output. 001: OC1REF is set high when CCR1 = CNT. 010: OC1REF is set low when CCR1 = CNT. 011: OC1REF is toggled when CCR1 = CNT. 100: OC1REF is fixed low (inactive). 101: OC1REF is fixed high (active).

Bit	Name	Attribute	Reset Value	Description
				110: PWM mode 1—In up-counting, OC1REF is set high when CNT < CCR1, otherwise set low; in down-counting, OC1REF is set low when CNT > CCR1, otherwise set high. 111: PWM mode 2—In up-counting, OC1REF is set low when CNT < CCR1, otherwise set high; in down-counting, OC1REF is set high when CNT > CCR1, otherwise set low.
3	OC1PE	RW	0x0	Output compare 1 preload enable: 0: The CCR1 preload register is disabled, and CCR1 can be written directly. 1: The CCR1 preload register is enabled, and read/write operations on CCR1 access the preload register, with content shifted to the shadow register only when an update event occurs.
2	OC1FE	RW	0x0	Output compare 1 fast enable: 0: Disable fast enable, the trigger input will not affect the comparison output. 1: Enable fast enable, the trigger input will immediately change OC1REF to the output corresponding to the compare match, regardless of the actual current comparison status. This function is only valid when the channel is configured in PWM1 or PWM2 mode, where PWM1 is fixed to output 0 and PWM2 is fixed to output 1.
1:0	CC1S	RW	0x0	Capture/compare 1 channel selection: 00: CC1 channel configured as output 01: CC1 channel configured as input, IC1 mapped to TI1 10: CC1 channel configured as input, IC1 mapped to TI2 11: CC1 channel configured as input, IC1 mapped to TRC Note: CC1S can only be written when the channel is disabled (CC1E = 0).

2. Input capture mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	IC2F	RW	0x0	Input capture 2 filter
11:10	IC2PSC	RW	0x0	Input capture 2 prescaler
9:8	CC2S	RW	0x0	Capture/compare channel 2 selection: 00: CC2 channel configured as output 01: CC2 channel configured as input, IC2 mapped to TI2 10: CC2 channel configured as input, IC2 mapped to TI1 11: CC2 channel configured as input, IC2 mapped to TRC Note: CC2S can only be written when the channel is disabled (CC2E = 0).
7:4	IC1F	RW	0x0	Input capture 1 filter: This bit-field defines the sampling frequency and filter length for TI1. 0000: No filter, sampling at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
3:2	IC1PSC	RW	0x0	Input capture 1 prescaler: 00: No prescaling

Bit	Name	Attribute	Reset Value	Description
				01: Capture occurs once every 2 event inputs. 10: Capture occurs once every 4 event inputs. 11: Capture occurs once every 8 event inputs. The IC1PSC register is reset when CC1E = 0.
1:0	CC1S	RW	0x0	Capture/compare channel 1 selection: 00: CC1 channel configured as output 01: CC1 channel configured as input, IC1 mapped to TI1 10: CC1 channel configured as input, IC1 mapped to TI2 11: CC1 channel configured as input, IC1 mapped to TRC Note: CC1S can only be written when the channel is disabled (CC1E = 0).

12.5.8 ATIMER Capture/Compare Mode Register 2 ATIMER_CCMR2 (Offset: 1CH)

This register is multiplexed for two different functions under output compare and input capture modes:

1. Output compare mode

Bit	Name	Attribute	Reset Value	Description
31:18	RSV	-	-	Reserved
17:16	OC4SEL	RW	0x0	OC4REF rising/falling edge pulse trigger selection: 00: No pulse trigger 01: Pulse triggered on rising edge of OC4REF 10: Pulse triggered on falling edge of OC4REF 11: Pulse triggered on both rising and falling edges of OC4REF
15	OC4CE	RW	0x0	Output compare 4 clear enable, refer to OC1CE
14:12	OC4M	RW	0x0	Output compare 4 mode configuration, refer to OC1M

Bit	Name	Attribute	Reset Value	Description
11	OC4PE	RW	0x0	Output compare 4 preload enable, refer to OC1PE
10	OC4FE	RW	0x0	Output compare 4 fast enable, refer to OC1FE
9:8	CC4S	RW	0x0	Capture/compare channel 4 selection: 00: CC4 channel configured as output 01: CC4 channel configured as input, IC4 mapped to TI4 10: CC4 channel configured as input, IC4 mapped to TI3 11: CC4 channel configured as input, IC4 mapped to TRC Note: CC4S can only be written when the channel is disabled (CC4E = 0).
7	OC3CE	RW	0x0	Output compare 4 clear enable: 0: OC4REF is not affected by ETRF. 1: OC4REF is automatically cleared when ETRF is high.
6:4	OC3M	RW	0x0	Output compare 3 mode: these bits define the behavior of the OC3REF signal. 000: The comparison result between the output compare register CCR3 and the counter CNT does not affect the output. 001: OC3REF is set high when CCR3 = CNT. 010: OC3REF is set low when CCR3 = CNT. 011: OC3REF is toggled when CCR3 = CNT. 100: OC3REF is fixed low (inactive). 101: OC3REF is fixed high (active). 110: PWM mode 1—In up-counting, OC3REF is set high when CNT < CCR3, otherwise set low; in down-counting, OC3REF is set low when CNT > CCR3, otherwise set high. 111: PWM mode 2—In up-counting, OC3REF is set low when CNT < CCR3, otherwise set high; in down-counting, OC3REF is set high when CNT > CCR3, otherwise set low.
3	OC3PE	RW	0x0	Output compare 3 preload enable: 0: The CCR3 preload register is disabled; and CCR3 can be written directly.

Bit	Name	Attribute	Reset Value	Description
				1: The CCR3 preload register is enabled, and read/write operations on CCR3 access the preload register, with content shifted to the shadow register only when an update event occurs.
2	OC3FE	RW	0x0	Output compare 3 fast enable: 0: Disable fast enable, the trigger input will not affect the comparison output. 1: Enable fast enable, the trigger input will immediately change OC3REF to the output corresponding to the compare match, regardless of the actual current comparison status. This function is only valid when the channel is configured in PWM1 or PWM2 mode, where PWM1 is fixed to output 0 and PWM2 is fixed to output 1.
1:0	CC3S	RW	0x0	Capture/compare channel 3 selection: 00: CC3 channel configured as output 01: CC3 channel configured as input, IC3 mapped to TI3 10: CC3 channel configured as input, IC3 mapped to TI4 11: CC3 channel configured as input, IC3 mapped to TRC Note: CC3S can only be written when the channel is disabled (CC3E = 0).

2. Input capture mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	IC4F	RW	0x0	Input capture 4 filter
11:10	IC4PSC	RW	0x0	Input capture 4 prescaler
9:8	CC4S	RW	0x0	Capture/compare channel 4 selection: 00: CC4 channel configured as output 01: CC4 channel configured as input, IC4 mapped to TI4 10: CC4 channel configured as input, IC4 mapped to TI3 11: CC4 channel configured as input, IC4 mapped to TRC

Bit	Name	Attribute	Reset Value	Description
				Note: CC4S can only be written when the channel is disabled (CC4E = 0).
7:4	IC3F	RW	0x0	Input capture 3 filter: This bit-field defines the sampling frequency and filter length for TI3. 0000: No filter, sampling at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
3:2	IC3PSC	RW	0x0	Input capture 3 prescaler: 00: No prescaling 01: Capture occurs once every 2 event inputs. 10: Capture occurs once every 4 event inputs. 11: Capture occurs once every 8 event inputs. Note: The IC3PSC register is reset when CC3E = 0.
1:0	CC3S	RW	0x0	Capture/compare channel 3 selection: 00: CC3 channel configured as output 01: CC3 channel configured as input, IC3 mapped to TI3 10: CC3 channel configured as input, IC3 mapped to TI4 11: CC3 channel configured as input, IC3 mapped to TRC Note: CC3S can only be written when the channel is disabled (CC3E = 0).

12.5.9 ATIMER Capture/Compare Enable Register ATIMER_CCER (Offset: 20H)

Bit	Name	Attribute	Reset Value	Description
31:18	RSV	-	-	Reserved
17	CC5P	RW	0x0	Capture/compare 5 output polarity: refer to CC1P
16	CC5E	RW	0x0	Capture/compare 5 output enable: refer to CC1E
15:14	RSV	-	-	Reserved
13	CC4P	RW	0x0	Capture/compare 4 output polarity: refer to CC1P
12	CC4E	RW	0x0	Capture/compare 4 output enable: refer to CC1E
11	CC3NP	RW	0x0	Capture/compare 3 output polarity: refer to CC1NP
10	CC3NE	RW	0x0	Capture/compare 3 output enable: refer to CC1NE
9	CC3P	RW	0x0	Capture/compare 3 output polarity: refer to CC1P
8	CC3E	RW	0x0	Capture/compare 3 output enable: refer to CC1E
7	CC2NP	RW	0x0	Capture/compare 2 output polarity: refer to CC1NP
6	CC2NE	RW	0x0	Capture/compare 2 output enable: refer to CC1NE
5	CC2P	RW	0x0	Capture/compare 2 output polarity: refer to CC1P
4	CC2E	RW	0x0	Capture/compare 2 output enable: refer to CC1E
3	CC1NP	RW	0x0	Capture/compare 1 complementary output polarity: 0: OC1N is active high. 1: OC1N is active low.
2	CC1NE	RW	0x0	Capture/compare 1 complementary output enable: 0: OC1N is invalid; the level of OC1N is determined by the MOE, OSSI, OSSR, OIS1, OIS1N and CC1E bits.
1	CC1P	RW	0x0	Capture/compare 1 output polarity: When CC1 channel is configured as output: 0: OC1 is active high. 1: OC1 is active low. When CC1 channel is configured as input: 0: Non-inverted mode: capture occurs on the rising edge of IC1. 1: Inverted mode: capture occurs on the falling edge of IC1.
0	CC1E	RW	0x0	Capture/compare 1 output enable:

Bit	Name	Attribute	Reset Value	Description
				When CC1 channel is configured as output: 0: OC1 is not active. 1: OC1 is active. When CC1 channel is configured as input: 0: Capture function is disabled. 1: Capture function is enabled.

The following table shows the correspondence between the control register and the status of the complementary output channels, where MOE is the global timer output enable bit, OSSI is the off_state selection bit in IDLE state (MOE = 0), and OSSR is the off_state selection bit in RUN state (MOE = 1).

Off-state:

Control Register					Output State	
MOE	OSSI	OSSR	CCxE	CCxNE	OCx Output State	OCxN Output State
1	X	0	0	0	Output disabled (not driven by ATIMER), OCx = 0, OCx_EN = 0	Output disabled (not driven by ATIMER), OCxN = 0, OCxN_EN = 0
		0	0	1	Output disabled (not driven by ATIMER), OCx = 0, OCx_EN = 0	OCxREF + Polarity OCxN = OCxREF xor CCxNP, OCxN_EN = 1
		0	1	0	OCxREF + Polarity OCx = OCxREF xor CCxP, OCx_EN = 1	Output disabled (not driven by ATIMER), OCx = 0, OCx_EN = 0
		0	1	1	OCREF + Polarity + dead-time, OCx_EN = 1	Inverted OCREF + Polarity + dead-time, OCxN_EN = 1
		1	0	0	Output disabled (not driven by ATIMER), OCx = CCxP, OCx_EN = 0	Output disabled (not driven by ATIMER), OCxN = CCxNP, OCxN_EN = 0
		1	0	1	Off-state (output enabled with inactive state) OCx = CCxP, OCx_EN = 1	OCxREF + Polarity OCxN = OCxREF xor CCxNP, OCxN_EN = 1

<table>
<tr><th colspan="5">Control Register</th><th colspan="2">Output State</th></tr>
<tr><th>MOE</th><th>OSSI</th><th>OSSR</th><th>CCxE</th><th>CCxNE</th><th>OCx Output State</th><th>OCxN Output State</th></tr>
<tr><td rowspan="2"></td><td rowspan="2"></td><td>1</td><td>1</td><td>0</td><td>OCxREF + Polarity
OCx = OCxREF xor CCxP,
OCx_EN = 1</td><td>Off-state (output enabled with inactive state)
OCxN = CCxNP,
OCxN_EN = 1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>OCREF + Polarity + dead-time, OCx_EN = 1</td><td>Complementary to OCREF (not OCREF) + Polarity + dead-time,
OCxN_EN = 1</td></tr>
<tr><td rowspan="7">0</td><td>0</td><td rowspan="7">X</td><td>0</td><td>0</td><td>Output disabled (not driven by ATIMER),
OCx = CCxP, OCx_EN = 0</td><td>Output disabled (not driven by ATIMER),
OCxN = CCxNP,
OCxN_EN = 0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="2" rowspan="3">Output disabled (not driven by ATIMER)
If no clock is provided: OCx = CCxP, OCx_EN = 0, OCxN = CCxNP, OCxN_EN = 0
If the clock is provided: OCx = OISx, OCxN = OISxN after a dead-time.</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Output disabled (not driven by ATIMER),
OCx = CCxP, OCx_EN = 0</td><td>Output disabled (not driven by ATIMER),
OCxN = CCxNP,
OCxN_EN = 0</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>1</td><td colspan="2" rowspan="3">Off-state (output enabled with inactive state)
If no clock is provided: OCx = CCxP, OCx_EN = 1, OCxN = CCxNP, OCxN_EN = 1
If the clock is provided: OCx = OISx, OCxN = OISxN after a dead-time.</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
</table>

12.5.10 ATIMER Counter Register ATIMER_CNT (Offset: 24H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CNT	RW	0x0	Counter value

12.5.11 ATIMER Prescaler Register ATIME_PSC (Offset: 28H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	PSC	RW	0x0	Counter clock (CK_CNT) prescaler value: $f_{CK\_CNT} = f_{CK\_PSC} / (PSC[15:0] + 1)$ This is a preload register whose contents are transferred into the shadow register at each update event. Note: The maximum PWM output supported is 30 MHz, which shall be taken into account in the configuration of PSC and ARR.

12.5.12 ATIMER Auto-reload Register ATIMER_ARR (Offset: 2CH)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	ARR	RW	0x0	Auto-reload value at counter overflow: This is a preload register whose contents are transferred into the shadow register at each update event.

12.5.13 ATIMER Repetition Counter Register ATIMER_RCR (Offset: 30H)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:0	REP	RW	0x0	Repetition counter value: When REP is not 0, it decrements each time an update condition occurs. When REP = 0, it triggers an update event.

12.5.14 ATIMER Capture/Compare Register 1 ATIMER_CCR1 (Offset: 34H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR1	RW	0x0	Capture/compare channel 1 register **If channel CC1 is configured as output:** This is a preload register whose contents are loaded into the shadow register and then compared with the counter to generate the OC1 output. **If channel CC1 is configured as input:** CCR1 stores the counter value at the time of the most recent input capture event, at which point CCR1 is read-only.

12.5.15 ATIMER Capture/Compare Register 2 ATIMER_CCR2 (Offset: 38H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR2	RW	0x0	Capture/compare channel 2 register **If channel CC2 is configured as output:** This is a preload register whose contents are loaded into the shadow register and then compare with the counter to generate the OC2 output. **If channel CC2 is configured as input:** CCR2 stores the counter value at the time of the most recent input capture event, at which point CCR2 is read-only.

12.5.16 ATIMER Capture/Compare Register 3 ATIMER_CCR3 (Offset: 3CH)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR3	RW	0x0	Capture/compare channel 3 register **If channel CC3 is configured as output:** This is a preload register whose contents are loaded into the shadow register and then compare with the counter to generate the OC3 output. **If channel CC3 is configured as input:** CCR3 stores the counter value at the time of the most recent input capture event, at which point CCR3 is read-only.

12.5.17 ATIMER Capture/Compare Register 4 ATIMER_CCR4 (Offset: 40H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR4	RW	0x0	Capture/compare channel 4 register **If channel CC4 is configured as output:** This is a preload register whose contents are loaded into the shadow register and then compared with the counter to generate the OC4 output. **If channel CC4 is configured as input:** CCR4 stores the counter value at the time of the most recent input capture event, at which point CCR4 is read-only.

12.5.18 ATIMER Break and Dead-time Control Register ATIMER_BDTR (Offset: 44H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	MOE	RW	0x0	Output enable for master control: This register controls the output enable for all channels. Independent output enable for each channel is also controlled by CCxE and CCxNE. MOE can be set by software or automatically set by hardware when AOE = 1. When entering DBG, software will keep MOE at 0, and a hardware reset is required to exit. When the brake input is valid, MOE is asynchronously cleared by hardware. 0: OC and OCN outputs are disabled (with specific IO output state determined by OSSI). 1: OC and OCN outputs are enabled (still subject to the state of CCxE and CCxNE for each channel).
14	AOE	RW	0x0	Automatic output enable: 0: MOE can only be set by software. 1: MOE can be set by software or automatically set by an update event.
13	BKP	RW	0x0	Break polarity: 0: Break input is active low. 1: Break input is active high.
12	BKE	RW	0x0	Break enable: 0: Break input is disabled. 1: Break input is enabled.
11	OSSR	RW	0x0	Output off-state selection in run state: This bit is valid for channels with complementary outputs enabled only when MOE = 1. 0: When the output channel is disabled, the output enables of OC and OCN driving GPIO are 0. When CCxE or CCxNE = 1, the single output enable of the corresponding OC or OCN driving GPIO will be 1.

Bit	Name	Attribute	Reset Value	Description
				1: When the output channel is disabled, the output enables of OC and OCN driving GPIO are 0. Once either CCxE or CCxNE is 1, all the output enables of OC and OCN driving GPIO will be 1.
10	OSSI	RW	0x0	Output off-state selection in idle state: This bit is valid for the output channel only when MOE = 0. 0: When the output channel is disabled, the output enables of OC and OCN driving GPIO are 0. 1: When the output channel is disabled, the output enables of OC and OCN driving GPIO are 0. Once either CCxE or CCxNE is 1, all the output enables of OC and OCN driving GPI0 will be 1.
9:8	LOCK	RW	0x0	Register write protection configuration: 00: No write protection 01: Protection level 1— DTG, OISx, OISxN, BKE, BKP and AOE cannot be overwritten. 10: Protection level 2— In addition to level 1, CCxP, CCxNP, OSSR and OSSI cannot be overwritten. 11: Protection level 3— In addition to level 2, OCxM and OCxPE cannot be overwritten when the corresponding channel is configured as output. Note: The LOCK register cannot be overwritten after being written with a value other than 00, and the write-protected register can only be overwritten after the ATIMER module is reset.
7:0	DTG	RW	0x0	Dead-time insertion, used to configure the dead-time length inserted for complementary outputs: DTG[7:5] = 0xx: DT = DTG[7:0] * t_{DTS} DTG[7:5] = 10x: DT = (64 + DTG[5:0]) * 2 * t_{DTS} DTG[7:5] = 110: DT = (32 + DTG[4:0]) * 8 * t_{DTS} DTG[7:5] = 111: DT = (32 + DTG[4:0]) * 16 * t_{DTS}

12.5.19 ATIMER DMA Control Register ATIMER_DCR (Offset: 48H)

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12:8	DBL	RW	0x0	DMA burst length: Reading from or writing to the ATIMER_DMAR register will trigger a burst DMA operation, with the burst length ranging from 1 to 18. 00000: Length = 1 00001: Length = 2 …… 10001: Length = 18 Others: Invalid values, writing is prohibited.
7:5	RSV	-	-	Reserved
4:0	DBA	RW	0x0	DMA base address, defining the offset address directed to the register: 00000: ATIMER_CR1 00001: ATIMER_CR2 00010: ATIMER_SMCR …… Note: If DBA + DBL exceeds the address range of ATIMER register, the actual burst transfer will stop automatically after reaching the highest address of ATIMER register, i.e., the burst length will be shortened.

12.5.20 ATIMER DMA Access Register ATIMER_DMAR (Offset: 4CH)

Bit	Name	Attribute	Reset Value	Description
31:0	DMAR	RW	0x20000	DMA burst access register: When using DMA burst transfer, set the DMA channel peripheral address to ATIM_DMAR, and ATIMER will generate multiple DMA requests based on the value of DBL.

12.5.21 ATIMER Capture/Compare Mode Register 3 ATIMER_CCMR3 (Offset: 50H)

Bit	Name	Attribute	Reset Value	Description
31:18	RSV	-	-	Reserved
17:16	OC5SEL	RW	0x0	OC5REF rising/falling edge pulse trigger selection: 00: No pulse trigger 01: Pulse triggered on rising edge of OC5REF 10: Pulse triggered on falling edge of OC5REF 11: Pulse triggered on both rising and falling edges of OC5REF
15:8	RSV	-	-	Reserved
7	OC5CE	RW	0x0	Output compare 5 clear enable: 0: OC5REF is not affected by ETRF. 1: OC5REF is automatically cleared when ETRF is high.
6:4	OC5M	RW	0x0	Output compare 5 mode: these bits define the behavior of the output reference signal OC5REF. 000: The comparison result between the output compare register CCR5 and the counter CNT does not affect the output. 001: OC5REF is set high when CCR5 = CNT. 010: OC5REF is set low when CCR5 = CNT. 011: OC5REF is toggled when CCR5 = CNT. 100: OC5REF is fixed low (inactive). 101: OC5REF is fixed high (active). 110: PWM mode 1—In up-counting, OC5REF is set high when CNT < CCR5, otherwise set low; in down-counting, OC5REF is set low when CNT > CCR5, otherwise set high. 111: PWM mode 2—In up-counting, OC5REF is set low when CNT < CCR5, otherwise set high; in down-counting, OC5REF is set high when CNT > CCR5, otherwise set low.

Bit	Name	Attribute	Reset Value	Description
3	OC5PE	RW	0x0	Output compare 5 preload enable: 0: The CCR5 preload register is disabled; and CCR5 can be written directly. 1: The CCR5 preload register is enabled, and read/write operations on CCR5 access the preload register, with content shifted to the shadow register only when an update event occurs.
2	OC5FE	RW	0x0	Output compare 5 fast enable: 0: Disable fast enable, the trigger input will not affect the comparison output. 1: Enable fast enable, the trigger input will immediately change OC5REF to the output corresponding to the compare match, regardless of the actual current comparison status. This function acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	RSV	-	-	Reserved

12.5.22 ATIMER Capture/Compare Register 5 ATIMER_CCR5 (Offset: 54H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR5	RW	0x0	Capture/compare channel 5 register: This is a preload register whose contents are loaded into the shadow register and then compared with the counter to generate the OC5 output.

12.5.23 ATIMER PWM Phase-shift Enable Register ATIMER_PMEN (Offset: 58H)

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved
2	PM3EN	RW	0x0	Channel 3 phase shift enable: 0: Phase shift function disabled 1: Phase shift function enabled
1	PM2EN	RW	0x0	Channel 2 phase shift enable: 0: Phase shift function disabled 1: Phase shift function enabled
0	PM1EN	RW	0x0	Channel 1 phase shift enable: 0: Phase shift function disabled 1: Phase shift function enabled

12.5.24 ATIMER Phase-shift Offset Register 1 ATIMER_PMC1 (Offset: 5CH)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	PMC1	RW	0x0	Compare channel 1 offset register

12.5.25 ATIMER Phase-shift Offset Register 2 ATIMER_PMC2 (Offset: 60H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	PMC2	RW	0x0	Compare channel 2 offset register

12.5.26 ATIMER Phase-shift Offset Register 3 ATIMER_PMC3 (Offset: 64H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	PMC3	RW	0x0	Compare channel 3 offset register

12.5.27 ATIMER Break Input Control Register ATIMER_BKCTL (Offset: 68H)

Bit	Name	Attribute	Reset Value	Description
31:15	RSV	-	-	Reserved
14:12	BKINSEL	RW	0x0	Pin break source selection signal: 000: BRK0 001: BRK1 010: BRK2 011: BRK3 100: BRK4 101: BRK5 110: BRK6 111: Undefined
11	RSV	-	-	Reserved
10	CMP0_BRKEN	RW	0x0	Comparator 0 output break signal enable: 0: Disable CMP0 break signal 1: Enable CMP0 break signal
9	ADC_BRKEN	RW	0x0	ADC analog watchdog break signal enable: 0: Disable ADC analog watchdog break signal 1: Enable ADC analog watchdog break signal
8	BRKGATE	RW	0x1	ATIMER_BRK pin input gating signal: 0: Disable ATIMER_BRK pin input gating signal 1: Enable ATIMER_BRK pin input gating signal
7:4	BRKF	RW	0x0	Break signal filter clock and length selection:

Bit	Name	Attribute	Reset Value	Description
				0000: No filter 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS}$ / 2, N = 6 0101: $f_{SAMPLING} = f_{DTS}$ / 2, N = 8 0110: $f_{SAMPLING} = f_{DTS}$ / 4, N = 6 0111: $f_{SAMPLING} = f_{DTS}$ / 4, N = 8 1000: $f_{SAMPLING} = f_{DTS}$ / 8, N = 6 1001: $f_{SAMPLING} = f_{DTS}$ / 8, N = 8 1010: $f_{SAMPLING} = f_{DTS}$ / 16, N = 5 1011: $f_{SAMPLING} = f_{DTS}$ / 16, N = 6 1100: $f_{SAMPLING} = f_{DTS}$ / 16, N = 8 1101: $f_{SAMPLING} = f_{DTS}$ / 32, N = 5 1110: $f_{SAMPLING} = f_{DTS}$ / 32, N = 6 1111: $f_{SAMPLING} = f_{DTS}$ / 32, N = 8
3	RSV	-	-	Reserved
2	CMP1_BRKEN	RW	0x0	Comparator 1 output break signal enable: 0: Disable CMP1 break signal 1: Enable CMP1 break signal
1	SVD_BRKEN	RW	0x0	LVD break enable: 0: Disable LVD break signal 0: Enable LVD break signal
0	CODE_BRKEN	RW	0x0	Hardfault break signal enable: 0: Disable Hardfault break signal 1: Enable Hardfault break signal

12.6 Operation Procedure

12.6.1 Timer Counting Mode

1. Configure the DIR bit in ATIMER_CR1 to set the counting direction.

2. Set the APRE bit in ATIMER_CR1 to 1 to enable auto-reload preload.

3. Configure ATIMER_PSC to set the prescaler value.
4. Configure ATIMER_ARR to set the auto-reload value.
5. Set ATIMER_RCR to 0 to disable repetitive counting.
6. Set the URS bit in ATIMER_CR1 to 1 to generate an update interrupt or DMA request only at counter overflow or underflow.
7. Set the UDIS bit in ATIMER_CR1 to 0 to enable the update event.
8. Set the UG bit in ATIMER_EGR to 1 to reinitialize the counter, update the shadow registers, and clear the prescaler counter.
9. Set the CEN bit in ATIMER_CR1 to 1 to enable the counter.
10. Set the UIE bit in ATIMER_DIER to 1 to enable the update interrupt.

12.6.2 PWM Mode

1. Configure the DIR bit in ATIMER_CR1 to set the counting direction.
2. Set the APRE bit in ATIMER_CR1 to 1 to enable auto-reload preload.
3. Configure ATIMER_PSC to set the prescaler value.
4. Configure ATIMER_ARR to set the auto-reload value.
5. Set ATIMER_RCR to 0 to disable repetitive counting.
6. Set the CCxS bit in ATIMER_CCMRx to 0 to configure channel x as an output.
7. Configure the OCxM bit in ATIMER_CCMRx to set the PWM mode 1/2.
8. Configure the CCxP bit in ATIMER_CCER to set the output polarity.
9. Set the CCxE bit in ATIMER_CCER to 1 to enable channel x output.
10. Set the main output enable bit MOE in ATIMER_BDTR to 1 to enable OC and OCN outputs.
11. Set the URS bit in ATIMER_CR1 to 1 to generate an update interrupt or DMA request only at counter overflow or underflow.

12. Set the UDIS bit in ATIMER_CR1 to 0 to enable the update event.
13. Set the UG bit in ATIMER_EGR to 1 to reinitialize the counter, update the shadow registers, and clear the prescaler counter.
14. Set the CEN bit in ATIMER_CR1 to 1 to enable the counter.
15. Set the UIE bit in ATIMER_DIER to 1 to enable the update interrupt.
16. Configure ATIM_CCRx to set the comparison value for channel x.

12.6.3 Input Capture Mode

1. Configure the DIR bit in ATIMER_CR1 to set the counting direction.
2. Set the APRE bit in ATIMER_CR1 to 1 to enable auto-reload preload.
3. Configure ATIMER_PSC to set the prescaler value.
4. Configure ATIMER_ARR to set the auto-reload value.
5. Set ATIMER_RCR to 0 to disable repetitive counting.
6. Configure the CCxS bit in ATIMER_CCMRx to set CCx channel as an input and map it as required.
7. Configure the CCxP and CCxNP bits in ATIMER_CCER to set the capture polarity.
8. Configure the ICxF bit in ATIMER_CCMRx to set the sampling frequency and filter length, generally set to 0.
9. Configure the ICxPSC bit in ATIMER_CCMRx to set the input capture prescaler.
10. Set the CCxE bit in ATIMER_CCER to 1 to enable the capture function.
11. Set the UG bit in ATIMER_EGR to 1 to reinitialize the counter, update the shadow registers, and clear the prescaler counter.
12. Set the CEN bit in ATIMER_CR1 to 1 to enable the counter.
13. Set the CCxIE bit in ATIMER_DIER to 1 to enable the channel x capture interrupt.

12.6.4 Complementary Output and Dead-time Insertion

t_{DTS} = 16.66 ns at 60 MHz

1. DT = (0–127) * 16.66 = 0–2115.82 ns, DTG[7:5] = 0xx: DT = DTG[7:0] * t_{DTS}
2. DT = (64 + (0–63)) * 2 * 16.66 = 2132.48–4231.64 ns, DTG[7:5] = 10x: DT = (64+DTG[5:0]) * 2 * t_{DTS}
3. DT = (32 + (0–31)) * 8 * 16.66 = 4264.96–8396.64 ns, DTG[7:5] = 110: DT = (32+DTG[4:0]) * 8 * t_{DTS}
4. DT = (32 + (0–31)) * 16 * 16.66 = 8529.92–16793.68 ns, DTG[7:5] = 111: DT = (32+DTG[4:0]) * 16 * t_{DTS}

Before initializing the PWM mode, add the following configurations:

1. Configure the DTG bit in ATIMER_BDTR to set the dead time length for complementary outputs.
2. Set the CCxNE bit in ATIMER_CCER to 1 to enable the complementary output.

12.6.5 Break Function

Before initializing the PWM mode, add the following configurations:

1. Configure the OSSR bit in ATIMER_BDTR to set the output state when the timer is stopped in run mode.
2. Configure the OSSI bit in ATIMER_BDTR to set the output state when the timer is stopped in idle mode.
3. Configure the BKP bit in ATIMER_BDTR to set the break polarity.
4. Configure the CCxP bit in ATIMER_CCER to set the OCx output polarity.
5. Configure the CCxNP bit in ATIMER_CCER to set the OCxN output polarity.

6. Configure the OISx bit in ATIMER_CR2 to set the idle output state of OCx.
7. Configure the OISxN bit in ATIMER_CR2 to set the idle output state of OCxN.
8. Configure the AOE bit in ATIMER_BDTR to set the method of setting the MOE bit in ATIMER_BDTR.
9. Set the BKE bit in ATIMER_BDTR to 1 to enable break input.

12.6.6 Encoder Interface Mode

1. Configure the DIR bit in ATIMER_CR1 to set the counting direction.
2. Set the APRE bit in ATIMER_CR1 to 1 to enable auto-reload preload.
3. Configure ATIMER_PSC to set the prescaler value.
4. Configure ATIMER_ARR to set the auto-reload value.
5. Set ATIMER_RCR to 0 to disable repetitive counting.
6. Set the CC1S bit in ATIMER_CCMR1 to 1 to configure channel CC1 as input, with IC1 mapped to TI1.
7. Set the CC2S bit in ATIMER_CCMR1 to 1 to configure channel CC2 as input, with IC2 mapped to TI2.
8. Configure the CC1P and CC1NP bits in ATIMER_CCER to set the capture polarity.
9. Configure the CC2P and CC2NP bits in ATIMER_CCER to set the capture polarity.
10. Configure the IC1F bit in ATIMER_CCMR1 to set the sampling frequency and filter length, generally set to 0.
11. Configure the IC2F bit in ATIMER_CCMR1 to set the sampling frequency and filter length, generally set to 0.
12. Configure the SMS bit in ATIMER_SMCR to set encoder mode 1/2/3.
13. Set the CC1E bit in ATIMER_CCER to 1 to enable capture function for channel 1.

14. Set the CC2E bit in ATIMER_CCER to 1 to enable capture function for channel 2.
15. Set the UG bit in ATIMER_EGR to 1 to reinitialize the counter, update the shadow registers, and clear the prescaler counter.
16. Set the CEN bit in ATIMER_CR1 to 1 to enable the counter.
17. Set the CC1IE bit in ATIMER_DIER to 1 to allow channel 1 capture interrupt.

12.6.7 DMA Mode

In output compare mode, the value in SRAM is transferred to the compare register of ATIMER via DMA.

1. Before setting the UG bit by software and enabling the counter in PWM mode, add the following configurations:
2. Configure the DBL bit in ATIMER_DCR to set the DMA burst length.
3. Configure the DBA bit in ATIMER_DCR to set the DMA base address. Generally, the base address here shall be the capture/compare register corresponding to the relevant compare channel.
4. Set the CCxDE bit in ATIMER_DIER to 1 to enable the CCx DMA request.
5. Set the CCDS bit in ATIMER_CR2 to 0 to enable the generation of CCxDMA request at CCx event.
6. For details on DMA controller configuration, please refer to Chapter DMA.
7. After enabling the DMA transfer, when the counter value equals the compare value, DMA will transfer the value from SRAM to the base address.

13 General-purpose Timer GTIMER0

13.1 Overview

The general-purpose timer/counter (GTIMER0) features a 32-bit auto-reload counter and a programmable prescaler, with its own independent interrupt. It serves various purposes, including measuring the pulse width of input signals (input capture) and generating output waveforms. The counter can count up, down or up/down with the counter value accessible by software at any time.

13.2 Main Features

- 32-bit up, down, up/down auto-reload counter
- 16-bit programmable prescaler allowing real-time adjustment of the counter clock division
- Flexible selection of counter clock source
- Channels available for input capture, output compare, PWM (edge- or center-aligned mode), and single-pulse output
- Supporting cascading between timers
- Interrupt generated on the following events:
  - Update interrupt: counter overflow/underflow
  - Input capture
  - Output compare
- Supporting incremental quadrature encoder and Hall sensor

13.3 GTIMER0 Block Diagram

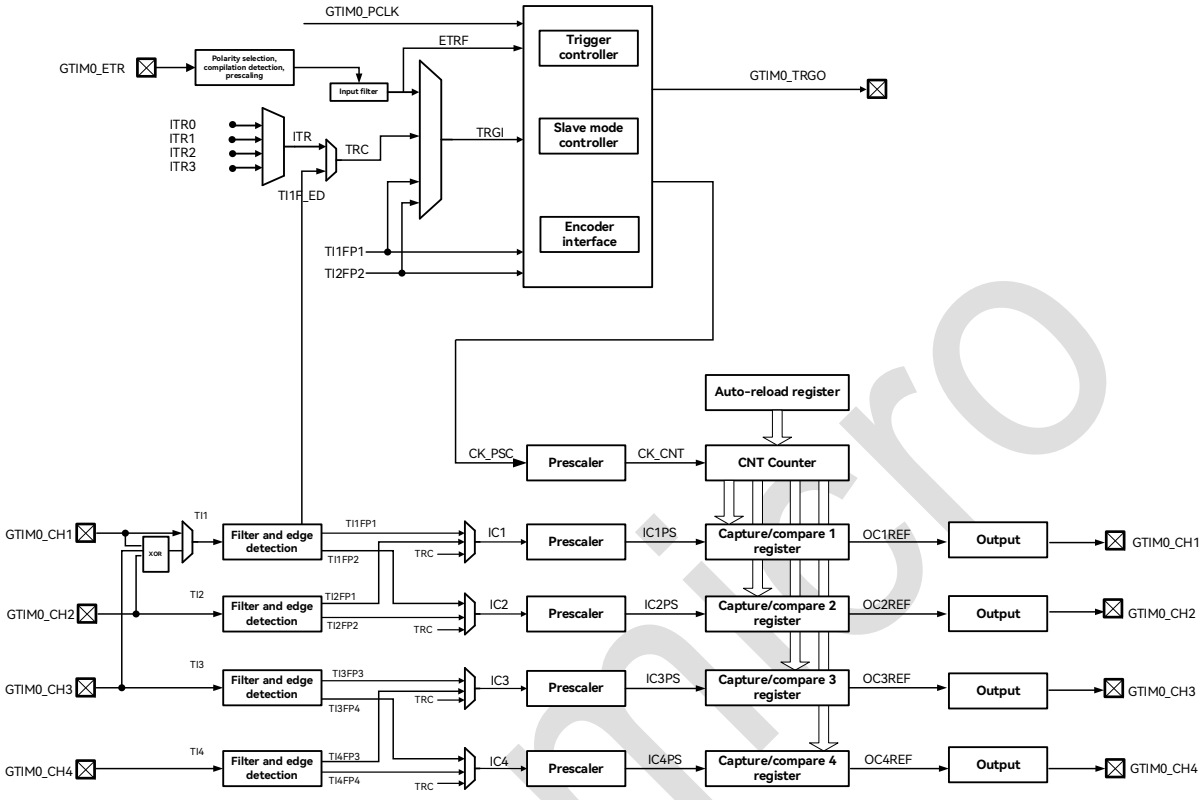


Figure 13-1: GTIMER0 Block Diagram

13.4 Functional Description

13.4.1 Timing Unit

The main components of GTIMER0 is a 32-bit counter and its associated auto-reload register. This counter can count up, down, or both up and down. Its clock is derived from a prescaler. The counter, auto-reload register and prescaler can be read and written by software even when the counter is running.

The timing unit consists of the following registers:

- Counter register (GTIMER_CNT)

- Prescaler register (GTIMER_PSC)
- Auto-reload register (GTIMER_ARR)

The GTIMER_ARR features a preload feature. According to the setting of the auto-reload preload enable bit (ARPE) in the GTIMER_CR1 register, the content of the preload register is transferred to the shadow register either immediately or upon each update event (UEV). An update event occurs when the counter reaches the overflow condition (underflow condition during down-counting) and when the UDIS bit in the GTIMER_CR1 register is 0. Software can also actively trigger an update of GTIMER_ARR by writing GTIMER_EGR[0] to 1.

The operating clock of the counter is driven by the divided clock generated by GTIMER_PSC. GTIMER_CNT starts counting only when the counter enable bit (CEN) is set.

GTIMER_PSC is a synchronous prescaler, which can divide the counter clock frequency by any value between 1 and 65536. It is a 16-bit counter controlled by a 16-bit register. Since this control register is equipped with a buffer, it can be modified on the fly. The new prescaler parameters take effect at the next update event.

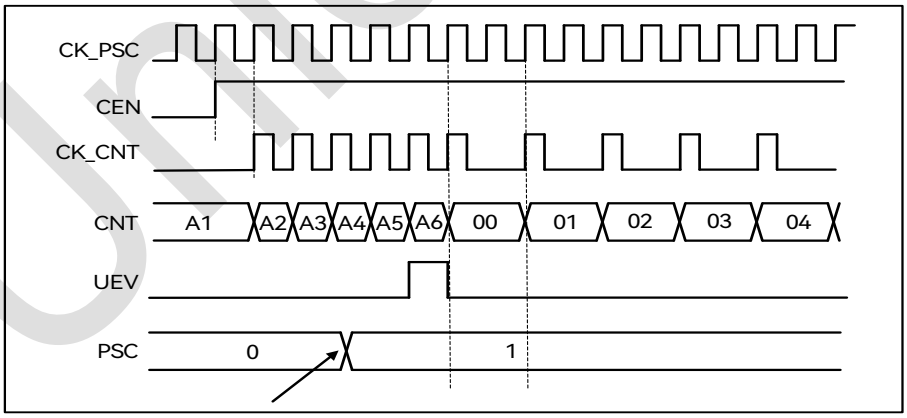


Figure 13-2: GTIMER Timing Diagram with Prescaler Division Changing from 1 to 2

13.4.2 Timer Operating Modes

The timer supports up-counting, down-counting and center-aligned counting modes.

13.4.2.1 Up-counting

In up-counting mode, the counter counts from 0 to the auto-reload value (the value in the GTIMER_ARR register), then restarts counting from 0, generating a counter overflow event.

Setting the UG bit in the GTIMER_EGR register (either through software or using the slave mode controller) can also generate an update event. Setting the UDIS bit in the GTIMER_CR1 register can disable the update event, preventing the shadow register from being updated when writing new values into the preload register. No update events will be generated until the UDIS bit is cleared. However, when an update event should occur, the counter will still be reset to 0, and the prescaler counter will also be reset to 0 (though the prescaler value remains unchanged).

If the URS bit in the GTIMER_CR1 register is set, setting the UG bit will generate an update event (UEV), but the hardware will not set the UIF flag.

When an update event occurs, the following registers are updated, and the UIF flag is set by hardware simultaneously:

- The shadow register of GTIMER_ARR is updated with the content of GTIMER_ARR.
- The shadow register of GTIMER_PSC is updated with the content of GTIMER_PSC.

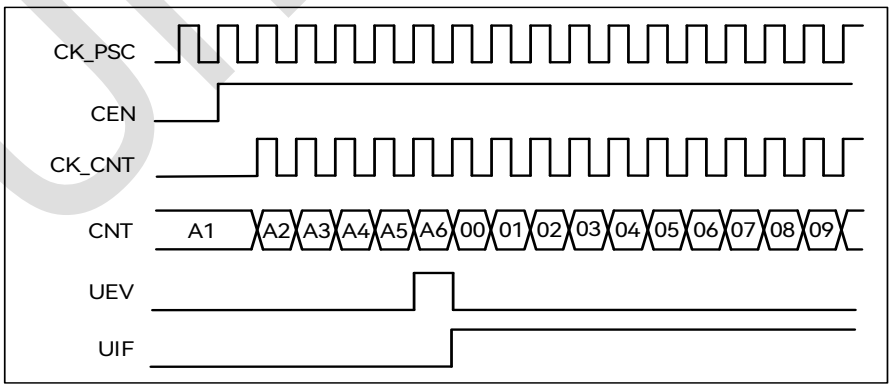


Figure 13-3: Up-counting Waveform

13.4.2.2 Down-counting

In down-counting mode, the counter counts down from the value in GTIMER_ARR to 0, then restarts counting from the value in GTIMER_ARR, generating a counter underflow event.

Setting the UG bit in the GTIMER_EGR register (either through software or using the slave mode controller can also generate an update event. Setting the UDIS bit in the GTIMER_CR1 register can disable the update event, preventing the shadow register from being updated when writing new values into the preload register. No update events will be generated until the UDIS bit is cleared. However, when an update event should occur, the counter will still be reset to 0, and the prescaler counter will also be reset to 0 (though the prescaler value remains unchanged).

If the URS bit in the GTIMER_CR1 register is set, setting the UG bit will generate an update event (UEV), but the hardware will not set the UIF flag.

When an update event occurs, the following registers are updated, and the UIF flag is set by hardware simultaneously:

- The shadow register of GTIMER_ARR is updated with the content of GTIMER_ARR.
- The shadow register of GTIMER_PSC is updated with the content of GTIMER_PSC.

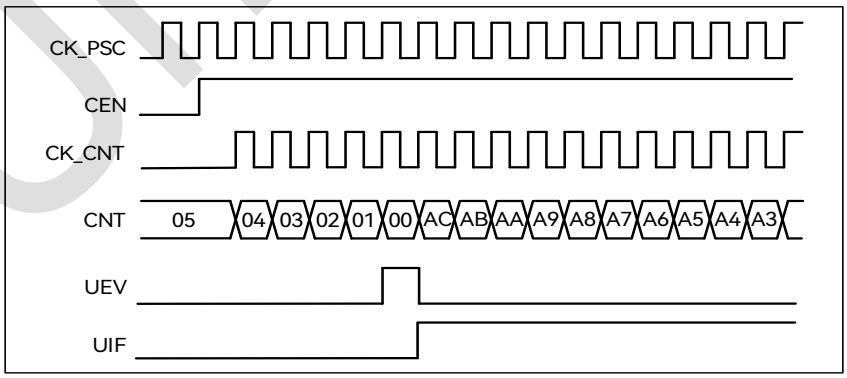


Figure 13-4: Down-counting Waveform

13.4.2.3 Center-aligned Counting

In center-aligned mode, the counter starts counting from 0 up to (ARR - 1), generating an overflow event; then starts counting from GTIMER_ARR down to 1, generating an underflow event, and restarts counting up from 0.

The GTIMER_CR1[6:5] register is used to enable the center-aligned mode and select the output compare mode therein:

- CMS = 00: Edge-aligned mode
- 01: Center-aligned mode 1, where the output compare interrupt flag is set only during down-counting.
- 10: Center-aligned mode 2, where the output compare interrupt flag is set only during up-counting.
- 11: Center-aligned mode 3, where the output compare interrupt flag is set during both up-counting and down-counting.
- In center-aligned mode, the DIR direction bit in the register cannot be rewritten by software, it is automatically updated by hardware as the counting direction changes, indicating the current counting direction. The following registers are updated on both overflow and underflow events of the counter:
- The shadow register of GTIMER_ARR is updated with the content of GTIMER_ARR.
- The shadow register of GTIMER_PSC is updated with the content of GTIMER_PSC.

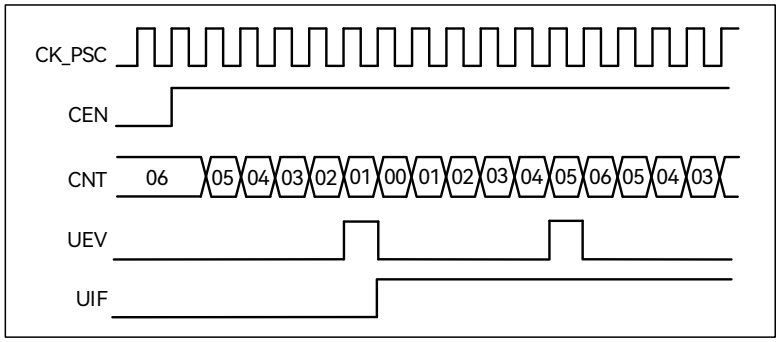


Figure 13-5: Center-aligned Counter Timing Diagram

13.4.3 Counter Operating Clock

The counter can operate with the following clocks:

- Internal clock APBCLK (CK_INT)
- External pin input clock GTIMER_CHx (x = 1, 2)
- External pin trigger input (GTIMER_ETR)
- Internal trigger (ITRx)

13.4.3.1 APBCLK

When APBCLK is selected as the clock source, setting the GTIMER to slave mode (SMS = 000) is prohibited. Register bits such as CEN, DIR and UG are all controlled by software. After the software operates the UG register, the frequency divider and counter value will be reinitialized.

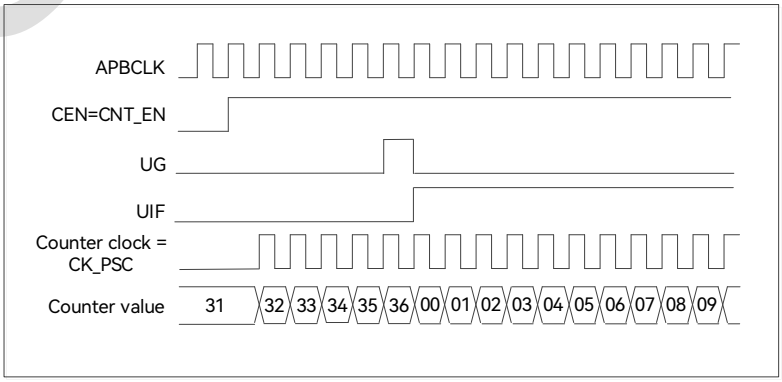


Figure 13-6: APBCLK Clock Source Mode with a Clock Prescaler Factor of 1

13.4.3.2 GTIMER_CHx (x = 1, 2) Pin Input: External Clock Mode 1

When configuring SMS = 111 in this mode, the input signal of the external pin GTIMER_CHx is directly used as the counting clock, with SMS configured to 111, and the counting edge can be selected as either the rising or falling edge.

In this mode, only the inputs of channels 1 and 2 can be used as clock inputs. Taking GTIMER_CH1 as an example, the configuration process is as follows:

1. Configure the corresponding pin as GTIMER_CH2 function in the IO module.
2. Configure GTIMER->CCMR1[9:8] = 01 to set channel 2 to detect the rising edge of TI2 input.
3. Configure GTIMER->CCMR1[15:12] to the desired filter value to select the input filter bandwidth (if no filter is required, keep CCMR1[15:12] = 0000).
4. Configure GTIMER->CCER[5] = 0 to set the rising edge polarity of TI2.
5. Configure GTIMER->SMCR[2:0] = 111 to select the timer external clock mode 1.
6. Configure GTIMER->SMCR[6:4] = 110 to select TI2 as the trigger input source.
7. Configure GTIMER-> CCER[0] = 1 to enable the channel.
8. Configure GTIMER-> CR1[0] = 1 to enable the counter.

13.4.3.3 GTIMER_ETR Pin Input: External Clock Mode 2

In this mode, the rising edge or falling edge of the input signal at the GTIMER_ETR pin is used for counting. The delay between the rising edge of ETR and the actual counter clock depends on the synchronization circuit at the ETRP signal terminal.

When using external clock mode 2, the GTIMER can still be configured in slave mode: for example, using the GTIMER_ETR input for counting while using the TRGO of another GTIMER as a trigger signal. When a trigger event occurs, the counter is reset and starts counting again.

13.4.4 Internal Trigger Signal (ITRx)

The GTIMER supports four internal trigger inputs, which can be used for counting triggers or internal signal capture. When used for internal signal capture, TS must be configured to 000–011 for selecting ITR0–ITR3, and CCxS must be configured to 11 for selecting TRC as the capture signal.

Each ITR input supports four internal signal extensions, configured by the TS register. Please refer to the table below for input signal sources:

Table 13-1: GTIMER0/1/2 Internal Signal Extension Table

Slave	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
GTIMER0	RCL32k	ATIMER_TRGO	COMP0_IN	COMP1_IN
GTIMER1	RCL32k	GTIMER0_TRGO	COMP0_IN	COMP1_IN
GTIMER2	RCL32k	GTIMER1_TRGO	COMP0_IN	COMP1_IN

13.4.5 Capture/Compare Channels

GTIMER1 includes 4 capture/compare channels, each of which consists of a capture/compare register GTIMER_CCR (including a shadow register), an input circuit (digital filter, multiplexer, and prescaler), and an output circuit (comparator and output control).

13.4.5.1 Capture Channel

The input circuit samples the corresponding TIx (x = 1...4) and generates a filtered signal TIxF (x = 1...4), which is then processed to generate the corresponding TIxFPx (x = 1...4) signal through edge detection and polarity selection. This signal can be used as a counting trigger source or a capture source, and it enters the capture channel after passing through the prescaler.

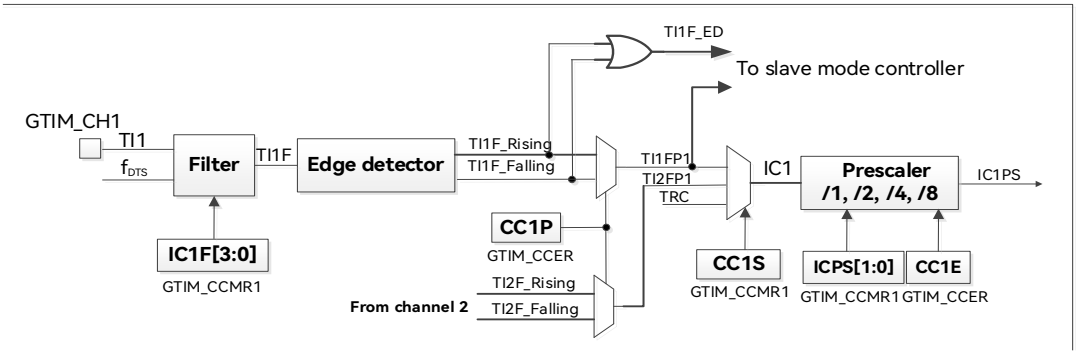


Figure 13-7: Capture/Compare Channel (Input Section of Channel 1)

13.4.5.2 Compare Channel

The output stage circuit generates an intermediate signal OCxREF (active high) as a reference, which serves as the reference input for the final output circuit.

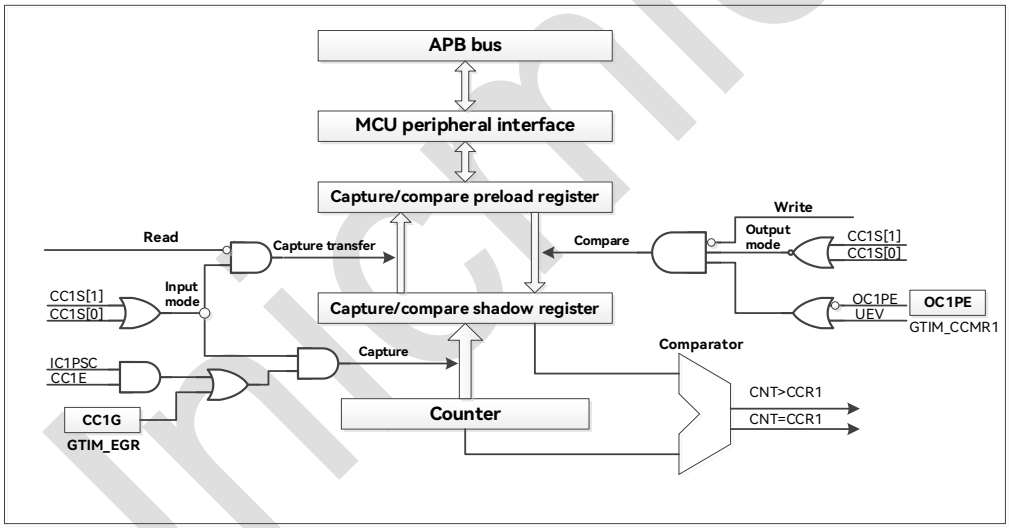


Figure 13-8: Main Circuit of Capture/Compare Channel 1

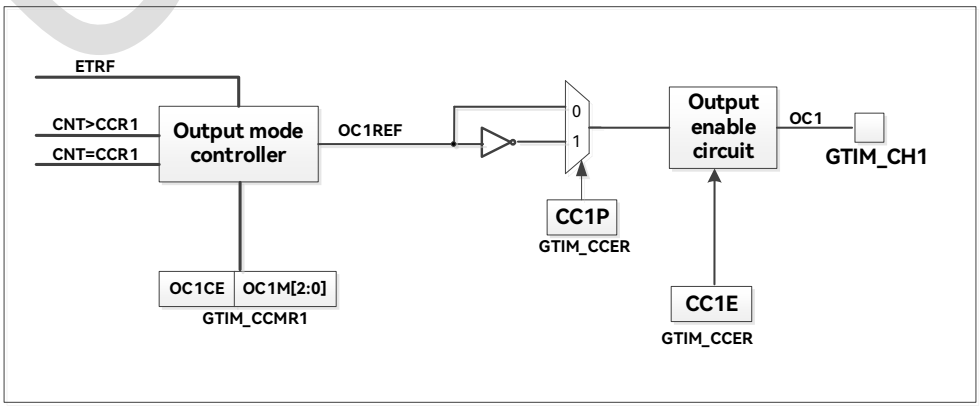


Figure 13-9: Output Section of Capture/Compare Channel

The capture/compare module consists of a preload register and a shadow register. Read and write operations only act on the preload register. In capture mode, capture occurs on the shadow register, and the captured value is then copied to the preload register. In compare mode, the content of the preload register is copied to the shadow register, and then the content of the shadow register is compared with the counter.

13.4.6 Input Capture Mode

In input capture mode, when the corresponding edge is detected on the ICx signal, the current value of the counter is latched into the capture/compare register (GTIMER_CCRx). When a capture event occurs, the corresponding CCxIF flag (in the GTIMER_SR register) is set to 1. If interrupts or DMA are enabled, an interrupt or DMA request will be generated. If the CCxIF flag is already high when a capture event occurs, the overcapture flag CCxOF (in the GTIMER_SR register) is set to 1. Writing to CCxIF or reading the capture data stored in the GTIMER_CCRx register can also clear CCxIF. Writing CCxOF = 1 can clear CCxOF.

The following example illustrates how to capture the counter value on the rising edge of TI1 input, with the steps as follows:

1. In GPIO module, configure the corresponding pin for GTIMER_CH1 function.
2. Select the input channel by setting GTIMER->CCMR1.CC1S = 01, mapping IC1 to TI1.
3. Select the active counting edge to be the rising edge for the T1 channel by setting GTIMER_CCER.CC1P = 0.
4. Configure the input filter time by setting GTIMER_CCMR1.IC1F.
5. Configure the input prescaler by setting GTIMER_CCMR1.IC1PS.
6. Set CC1E = 1 in the GTIMER_CCER register to enable capturing the counter value into the capture register.

7. Enable the corresponding interrupt or DMA request as required.

8. Enable the channel by setting GTIMER_CCER[0] = 1.

13.4.7 PWM Input Mode

This mode is a special case of input capture mode, where the pulse width of a PWM signal can be captured using two channels in cooperation.

To implement PWM input capture, the configuration process is as follows:

1. In GPIO module, configure the corresponding pin for GTIMER_CH1 function.

2. Select the valid input for GTIMER_CCR1 by setting CC1S = 01 (selecting TI1) in the GTIMER_CCMR1 register.

3. Select the valid polarity for TI1FP1 by setting CC1P = 0 (rising edge active).

4. Select the valid input for GTIMER_CCR2 by setting CC2S = 10 (selecting TI1) in the GTIMER_CCMR1 register.

5. Select the valid polarity for TI1FP2 by setting CC2P = 1 (falling edge active).

6. Select the valid trigger input signal by setting TS = 101 in the GTIMER_SMCR register.

7. Configure the slave mode controller to reset mode by setting SMS = 100 in the GTIMER_SMCR register.

8. Enable capture by setting CC1E = 1 and CC2E = 1 in the GTIMER_CCER register.

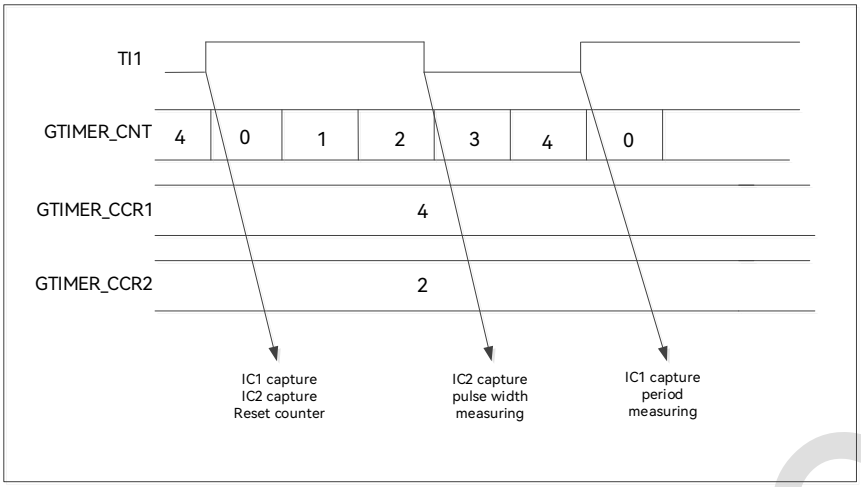


Figure 13-10: PWM Input Capture

13.4.8 Software-forced Output

In forced output mode, the output compare signals (OCxREF and the corresponding OCx/OCxN) can be directly forced to an active or inactive state by software, independent of the comparison result between the output compare register and the counter.

Setting the corresponding OCxM = 101 in the GTIMER_CCMRx register forces the output compare signal to an active state. In this case, OCxREF is forced to a high level, while OCx receives a signal with the opposite polarity of CCxP. For example: if CCxP = 0 (OCx active high), OCx is forced to a high level. Setting OCxM = 100 in the GTIMER_CCMRx register forces OCxREF to a low level.

13.4.9 Output Compare Mode

In output compare mode, when a match is found between GTIMER_CCR and GTIMER_CNT, OCxREF can be set to active, inactive or toggle on match. At the same time, the interrupt flag is also set and DMA requests can be sent. The output compare can also be used to output a pulse signal of a specific width (in single-pulse mode).

Operation procedure:

1. Select the counting clock (internal, external, prescaler, etc.).
2. Set the values of GTIMER_ARR and GTIMER_CCR registers.
3. Enable corresponding interrupts and DMA request as required.
4. Select the output mode, for example:
   - Set OCxM = 011 to toggle the OCx output pin when the counter matches CCRx.
   - Set OCxPE = 0 to disable the preload register.
   - Set CCxP = 0 to select the active-high polarity.
   - Set CCxE = 1 to enable output.

Set CEN = 1 to enable the counter.

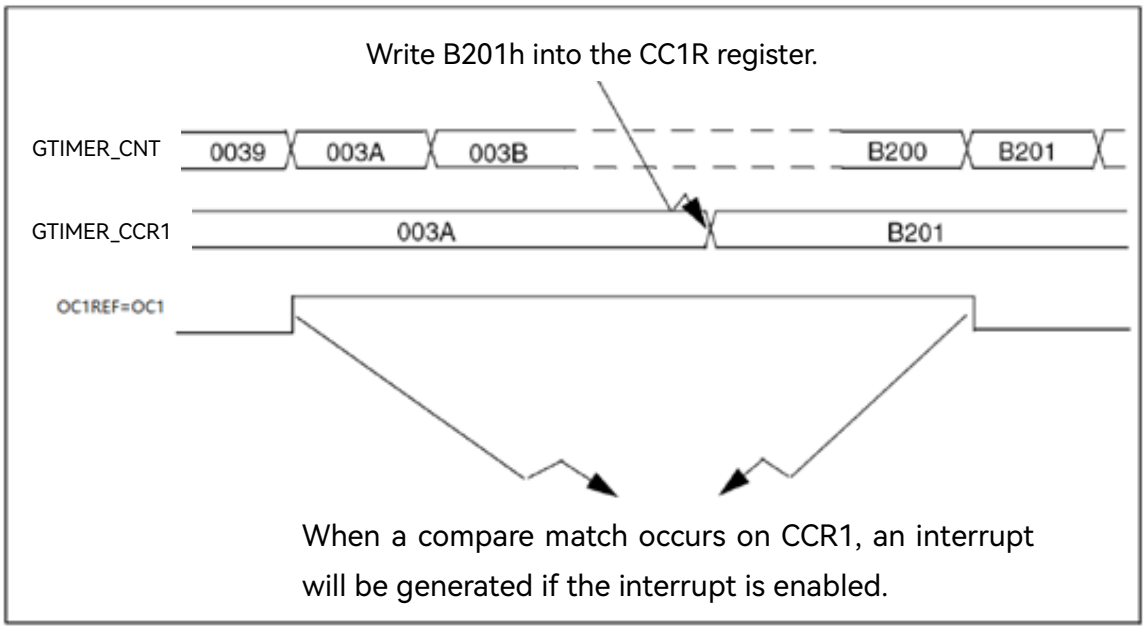


Figure 13-11: Output Compare Mode, Toggling OC1

If the preload function is disabled, the GTIMER_CCR register can be rewritten by software at any time to achieve real-time control of the output waveform. If the preload function is enabled, the shadow register of GTIMER_CCR will only be updated with the content of the preload register at the next update event.

13.4.10 PWM Output

The PWM mode can generate pulse-width modulation signals with the period determined by the GTIMER_ARR register and the duty cycle determined by the GTIMER_CCR register. The polarity of the output signal can be configured via the CCxP bit in the register. In PWM mode, GTIMER_CNT and GTIMER_CCR are compared in real time. The counter supports edge-aligned and center-aligned counting modes; accordingly, the PWM output also supports these two aligned modes.

13.4.10.1 PWM Edge-aligned Mode

In the case of up-counting, when configured in PWM mode 1, the OCxREF signal is high when GTIMER_CNT < GTIMER_CCR, and low otherwise. And OCxREF will be fixed at 1 if GTIMER_CCR > GTIMER_ARR, and fixed at 0 if CCR is 0.

In the case of down-counting, the definition of OCxREF level (high/low) is the same as that in up-counting mode.

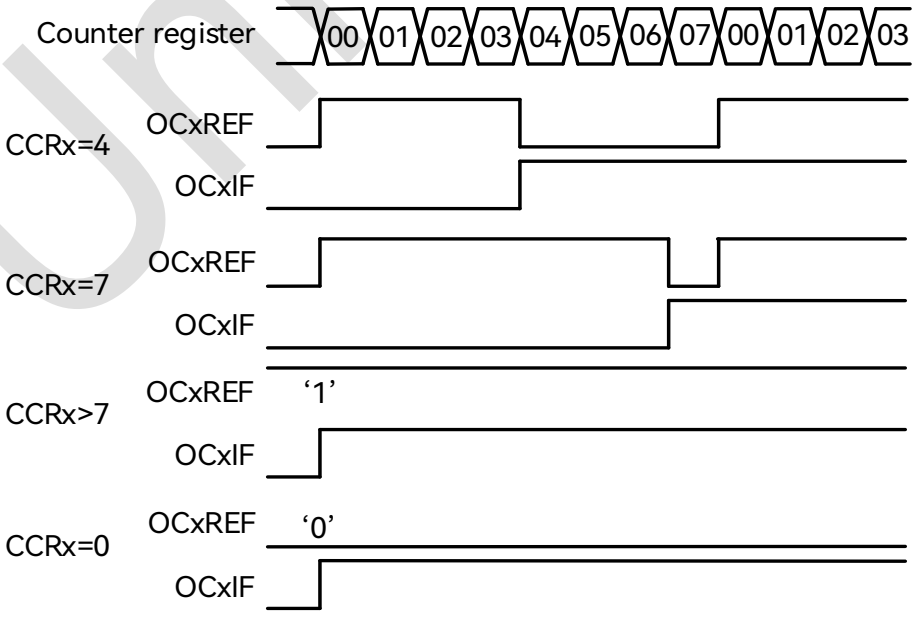


Figure 13-12: Edge-aligned PWM Waveform (ARR = 7)

13.4.10.2 PWM Center-aligned Mode

The definition of OCxREF level is the same as that in edge-aligned mode.

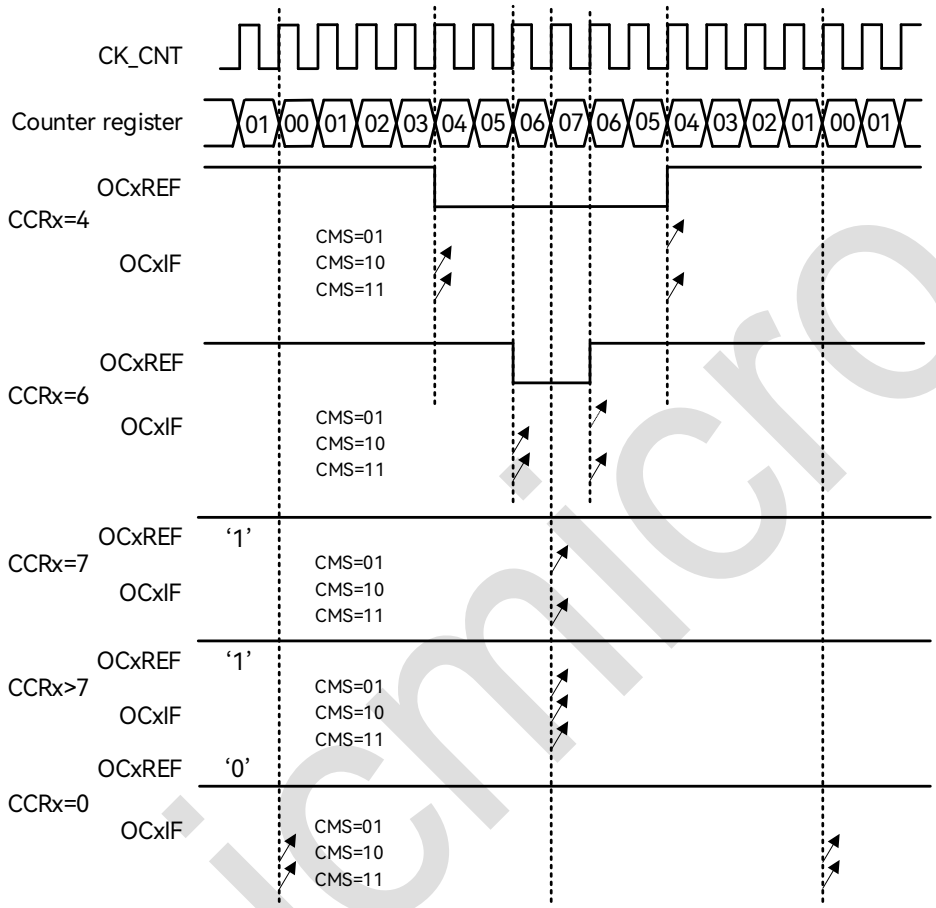


Figure 13-13: Center-aligned PWM Waveform (APR = 7)

When start counting in center-aligned mode, the initial counting direction is determined by the DIR bit in the register, and in the subsequent process, the DIR bit is directly controlled by hardware. The safest way to use center-aligned mode is to generate an update by setting the UG bit in the register just before starting the counter and not to overwrite the counter while it is running.

13.4.11 Output Compare Fast Enable Mode

When the fast enable mode is enabled (CCMR1[2,10], CCMR2[2,10]), the comparison output result of GTIMER is affected by the TRGI signal. The selection of TRGI signal is set through the

SMCR[4:6] register. Taking channel 1 as an example:

Assuming CC1P is 0:

- After the TRGI signal arrives, if PWM mode 1 is adopted, regardless of the original level of PWM, GTIMER_CH1 is forced to a low level (0), and returns to normal once the count reaches GTIMER_ARR.
- After the TRGI signal arrives, if PWM mode 2 is adopted, regardless of the original level of PWM, GTIMER_CH1 is forced to a low level (1), and returns to normal once the count reaches GTIMER_ARR.

Assuming CC1P is 1:

- After the TRGI signal arrives, if PWM mode 1 is adopted, regardless of the original level of PWM, GTIMER_CH1 is forced to a low level (1), and returns to normal once the count reaches GTIMER_ARR.
- After the TRGI signal arrives, if PWM mode 2 is adopted, regardless of the original level of PWM, GTIMER_CH1 is forced to a low level (0), and returns to normal once the count reaches GTIMER_ARR.

13.4.12 Single-pulse Output

Single-pulse output is a special case of the compare output mode, allowing users to output a pulse signal with a programmable width after a programmable delay following the occurrence of a specified event.

Unlike other output modes, the counter will automatically stop at the next update event. A pulse can be correctly generated only if the initial values of GTIMER_CCR and GTIMER_CNT are different. In up-counting, it is required that GTIMER_CNT < GTIMER_CCR ≤ GTIMER_ARR; while in down-counting, it is required that GTIMER_CNT > GTIMER_CCR.

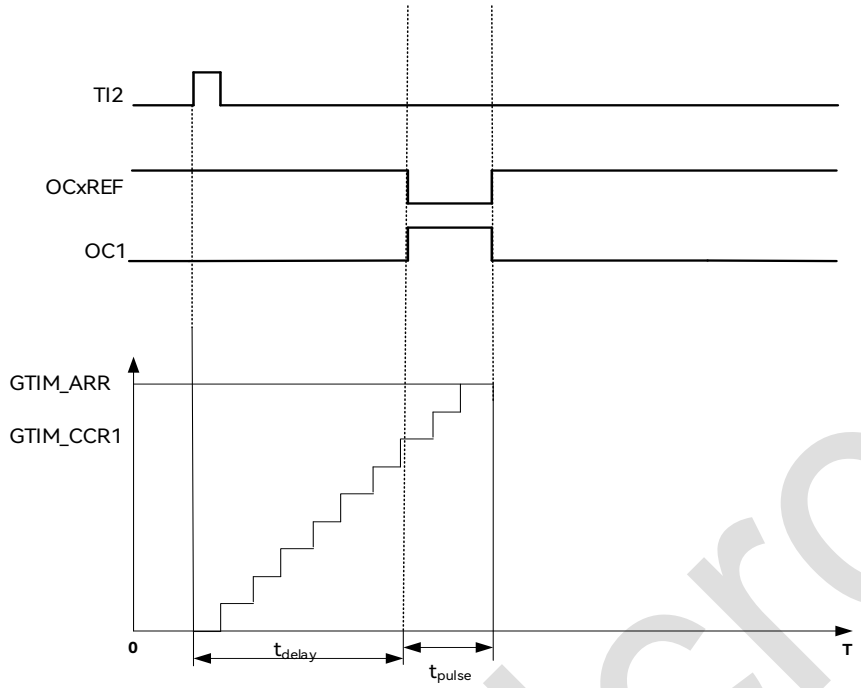


Figure 13-14: Example of Single-pulse Mode

The figure above shows that TI2 input is used as the counter trigger signal. When the counter value equals GTIMER_CCR, OCxREF outputs a low level. When the counter reaches GTIMER_ARR, OCxREF returns to a high level, and the counter returns to 0 and stops counting.

Configuration for realizing the above functions with TI2 as input trigger:

- In GPIO module, configure the corresponding pin for the GTIMER_CH2 function.
- Disable the channel by setting GTIMER->CCER.CC2E = 0 to ensure the success of subsequent configurations.
- Select the input channel by setting GTIMER->CCMR1.CC2S = 01.
- Select the active counting edge by setting GTIMER->CCER.CC2P = 0.
- Select TI2FP2 as the trigger input source by setting GTIMER->SMCR.TS[2:0] = 110.
- Set the slave mode controller to trigger mode by setting GTIMER->SMCR.SMS[2:0] = 110, with TI2FP2 for activating the counter.
- Enable the channel by setting GTIMER-> CCER.CC2E= = 1 to enable the channel.

Configuration for realizing the above functions with OC1 as output:

- In GPIO module, configure the corresponding pin for the GTIMER_CH1 function.
- Disable the channel by setting GTIMER->CCER.CC1E = 0 to ensure the success of subsequent configurations.
- Configure the output channel by setting GTIMER->CCMR1.CC1S = 00.
- Select the active counting edge by setting GTIMER->CCMR1.OC1M = 111, for PWM mode 2.
- Enable the channel by setting GTIMER->CCER.CC1E = 1.

Special settings for generating a single-pulse waveform base:

- t_{delay} is determined by the value of GTIMER->CCR1.
- t_{pulse} is determined by the difference between GTIMER->ARR and GTIMER->CCR1 (calculated as GTIMER->ARR - GTIMER->CCR1).
- Configure to single-pulse mode by setting GTIMER_CR1.OPM = 1.

13.4.13 Clearing OCxREF via External Event

OCxREF is active at high level, and it can be directly pulled down until the next update event by applying a high level to the external GTIMER_ETR pin. This function is only effective in output compare and PWM modes, and does not work in software force mode. To enable this function, OCxCE must be set to 1.

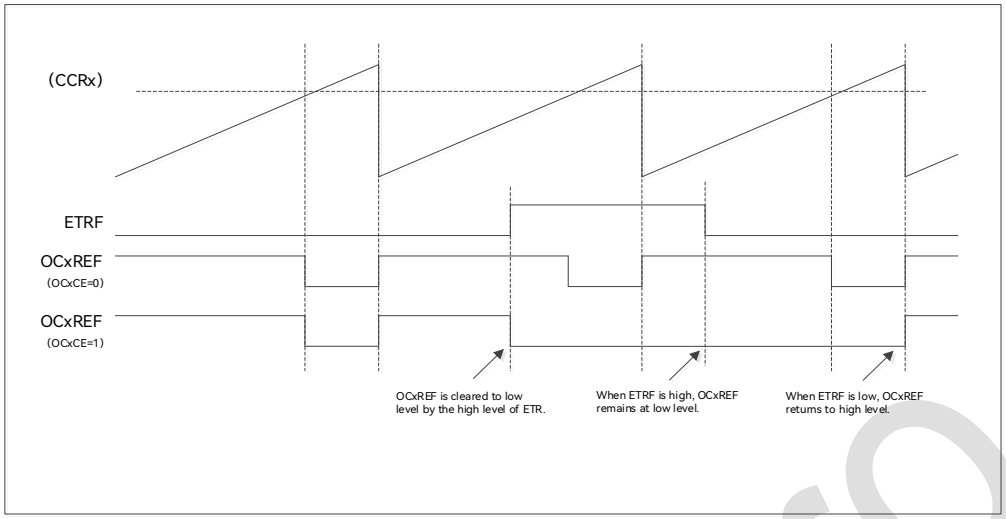


Figure 13-15: GTIMER_ETR Signal Clearing OCxREF of GTIMER

13.4.14 Encoder Interface Mode

The encoder interface mode involves two external input signals. GTIMER determines whether to increment or decrement the counter value based on the edge of one signal relative to the level of the other signal. The following table shows the relationship between the counting modes and the two input signals:

Table 13-2: Encoder Counting Methods and Input Signals

Counting Signal	TI2 Level When Counting on TI1/ TI1 Level When Counting on TI2	TI1 Signal		TI2 Signal	
		Rising Edge	Falling Edge	Rising Edge	Falling Edge
Counting on TI1	H	Decrement	Increment	No count	No count
	L	Increment	Decrement	No count	No count
Counting on TI2	H	No count	No count	Increment	Decrement
	L	No count	No count	Decrement	Increment
Counting on TI1 and TI2	H	Decrement	Increment	Increment	Decrement
	L	Increment	Decrement	Decrement	Increment

For example, when the counter counts using TI1 as the clock: if TI2 is sampled as high at the rising edge of TI1, the counter will decrement; if TI2 is sampled as high at the falling edge of TI1, the counter will increment.

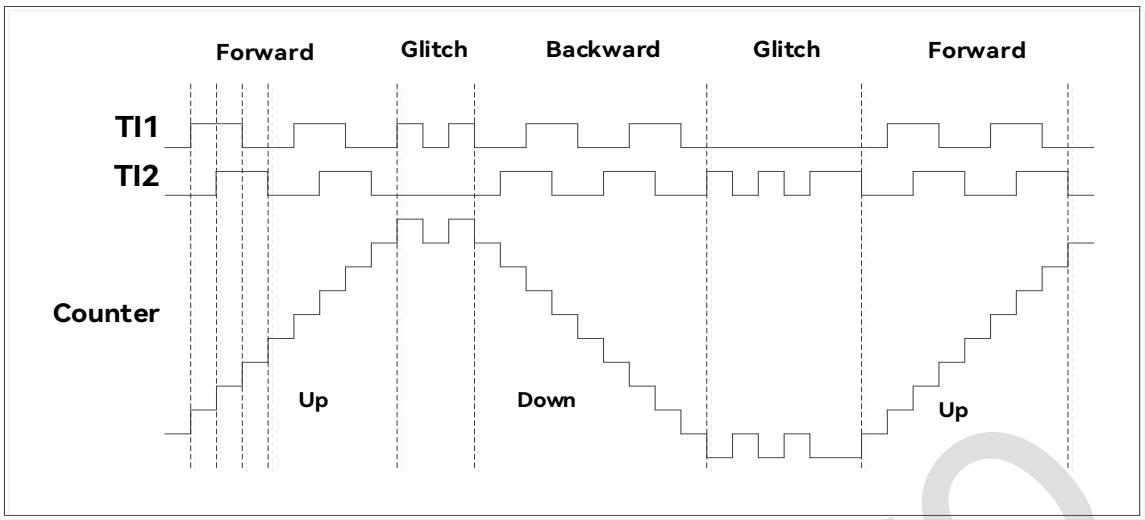


Figure 13-16: Example of Counter Operation in Encoder Mode

The input channels in encoder mode shall be configured as follows:

- In GPIO module, configure the corresponding pins for GTIMER_CH1 and GTIMER_CH2 functions.
- Disable the channel by setting GTIMER->CCER.CC1E = 0 and GTIMER->CCER.CC2E = 0 to ensure the success of subsequent channel configurations.
- Select the input channel by setting GTIMER->CCMR1.CC1S = 01 and GTIMER->CCMR1.CC2S = 01.
- Select the active counting edge by setting GTIMER->CCER.CC1P = 0 and GTIMER->CCER.CC2P = 0.
- Set the slave mode controller to encoder mode 3 by setting GTIMER->SMCR.SMS[2:0] = 011.

13.4.15 GTIMER Slave Mode

When GTIMER is used as a slave (triggered by an external event), it can be configured to operate in three modes: reset mode, gated mode, and trigger mode.

13.4.15.1 Reset Mode

In this mode, all the preload registers in GTIMER will be reinitialized due to external input events, and the counter will restart counting from 0. The counter counts normally until the rising edge of external TI1 input occurs, at which time the counter is cleared and then restarts counting. The configuration process is as follows:

1. In GPIO module, configure the corresponding pin for GTIMER_CH1 function.
2. Disable the channel by setting GTIMER_CCER.CC1E = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting GTIMER_CCMR1.CC1S = 01.
4. Select the active counting edge by setting GTIMER_CCER.CC1P = 0.
5. Select TI1FP1 as the trigger input signal (TRGI) by setting GTIMER_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to reset mode by setting GTIMER_SMCR.SMS[2:0] = 100.
7. Enable the channel by setting GTIMER_CCER.CC1E = 1.
8. Enable the counter by setting GTIMER_CR1.CEN = 1.

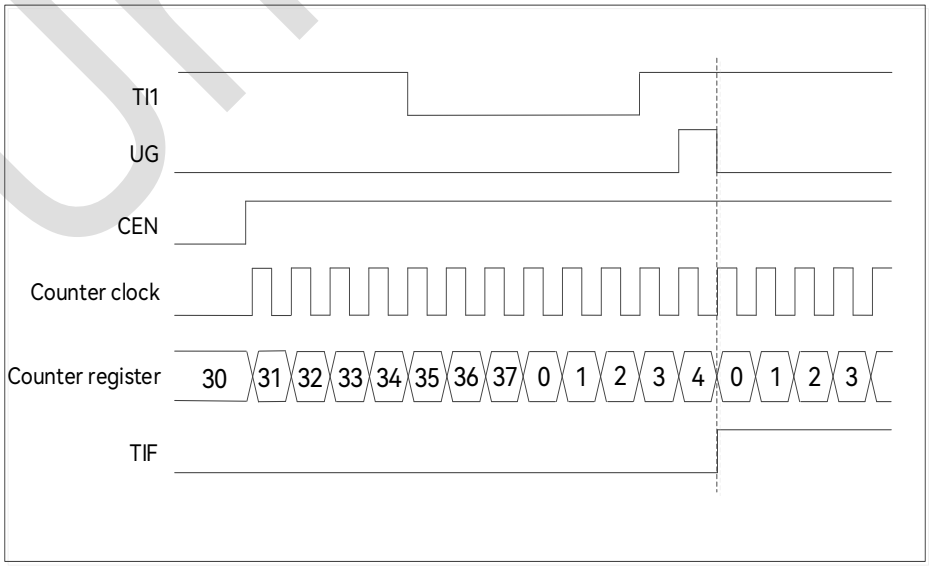


Figure 13-17: Timing Diagram in Reset Mode

13.4.15.2 Gated Mode

In gated mode, the counter operates only when the input signal is at a specific level. Level transitions that cause the counter to start or stop counting will trigger an interrupt flag. The timing configuration process is as follows:

1. In GPIO module, configure the corresponding pin for GTIMER_CH1 function.
2. Disable the channel by setting GTIMER_CCER.CC1E = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting GTIMER_CCMR1.CC1S = 01.
4. Select the active counting edge by setting GTIMER_CCER.CC1P = 1.
5. Select TI1FP1 as the trigger input signal (TRGI) by setting GTIMER_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to gated mode by setting GTIMER_SMCR.SMS[2:0] = 101.
7. Enable the channel by setting GTIMER_CCER.CC1E = 1.
8. Enable the counter by setting GTIMER_CR1.CEN = 1.

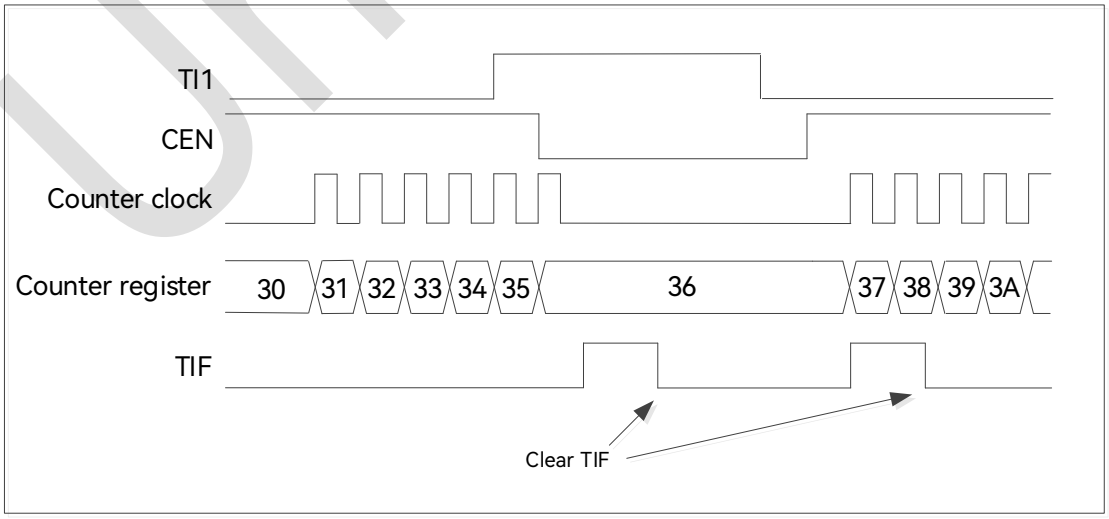


Figure 13-18: Timing Diagram in Gated Mode

13.4.15.3 Trigger Mode

The counter starts counting only after a specific external input event occurs. The timing configuration process is as follows:

1. In GPIO module, configure the corresponding pin for GTIMER_CH1 function.
2. Disable the channel by setting GTIMER->CCER.CC1E = 0 to ensure the success of subsequent channel configurations.
3. Select the input channel by setting GTIMER->CCMR1.CC1S = 01.
4. Select the active counting edge by setting GTIMER->CCER.CC1P = 0.
5. Select TI1FP1 as the trigger input source by setting GTIMER->SMCR.TS[2:0] = 101.
6. Set the slave mode controller to trigger mode by setting GTIMER->SMCR.SMS[2:0] = 110.
7. Enable the channel by setting GTIMER->CCER.CC1E = 1.

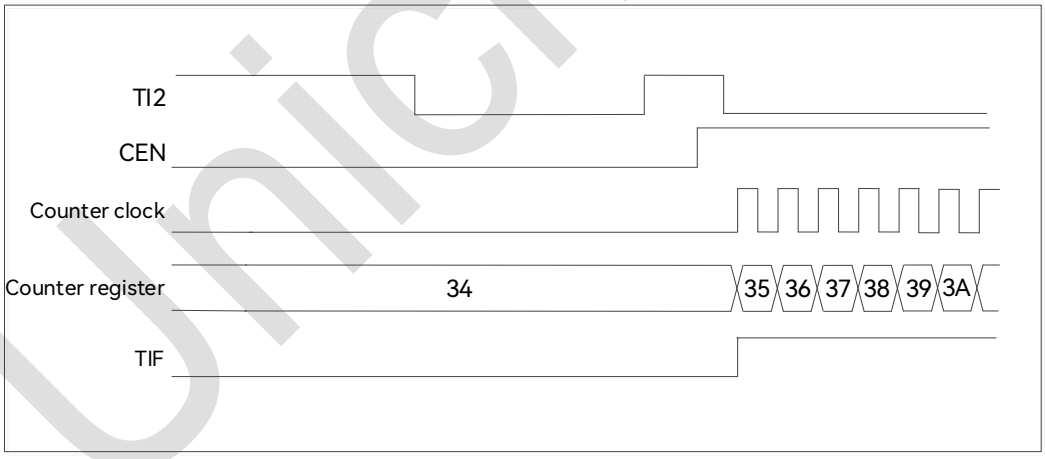


Figure 13-19: Timing Diagram in Trigger Mode

13.4.16 Timer Synchronization

Timers can be cascaded via trigger events to achieve synchronization or cascading. A general-purpose timer can be triggered to count by the outputs of two other general-purpose timers or

advanced timers. Meanwhile, the trigger output of a general-purpose timer can be connected to the internal trigger input of another timer. When a timer is in master mode, it can perform operations such as resetting, starting, stopping, or providing a clock for the counter of another timer in slave mode.

13.4.17 DMA Burst

GTIMER supports DMA and DMA-burst access. It can be configured to trigger a DMA request when a specific event occurs—this allows writing capture results from CCR to RAM, or writing the contents of one or more registers from RAM to the preload register of GTIMER.

DMA-burst allows for multiple successive DMA requests triggered by a single event, primarily for the purpose of continuously updating the contents of multiple registers after an event occurs, thus facilitating dynamic real-time adjustments to output waveforms and other functions.

The DMA controller must point the peripheral target address to a virtual register GTIMER_DMAR. When a specific timer event occurs, GTIMER will continuously issue multiple DMA requests. Each DMA write operation to GTIMER_DMAR will be redirected by GTIMER to the actual functional register.

The DBL register is used to set the DMA burst length, and the DBA register is used to set the base address (relative to the offset of GTIMER_CR) for DMA access to GTIMER.

13.4.18 Input XOR Function

The input signals from channels 1 to 3 can be XORed together and then connected to the input of the filtering and edge detection circuit for input capture or triggering of channel 1. The TI1S bit in the GTIMER_CR2 register is used to select whether the input of channel 1 comes from the XOR of the three channel inputs.

13.4.19 Debug Mode

When the Cortex-M0 enters debug mode, the timer can either stop or continue working, depending on the behavior defined by the STOP_EN register.

13.5 Register Description

13.5.1 GTIMER Registers

GTIMER0 register base address: 0x4000C000

Table 13-3: List of GTIMER Registers

Offset Address	Register	Description
0x00	GTIMER_CR1	GTIMER control register 1
0x04	GTIMER_CR2	GTIMER control register 2
0x08	GTIMER_SMCR	GTIMER slave mode control register
0x0C	GTIMER_DIER	GTIMER interrupt and DMA enable register
0x10	GTIMER_SR	GTIMER status register
0x14	GTIMER_EGR	GTIMER event generation register
0x18	GTIMER_CCMR1	GTIMER capture/compare mode register 1
0x1C	GTIMER_CCMR2	GTIMER capture/compare mode register 2
0x20	GTIMER_CCER	GTIMER capture/compare enable register
0x24	GTIMER_CNT	GTIMER counter register
0x28	GTIMER_PSC	GTIMER prescaler register
0x2C	GTIMER_ARR	GTIMER auto-reload register
0x34	GTIMER_CCR1	GTIMER capture/compare register 1
0x38	GTIMER_CCR2	GTIMER capture/compare register 2
0x3C	GTIMER_CCR3	GTIMER capture/compare register 3
0x40	GTIMER_CCR4	GTIMER capture/compare register 4
0x48	GTIMER_DCR	GTIMER DMA control register
0x4C	GTIMER_DMAR	GTIMER DMA access register

13.5.2 Control Register 1 GTIMER_CR1 (Offset: 00H)

Bit	Name	Attribute	Reset Value	Description
31:11	RSV	-	-	Reserved
10	STOP_EN	RW	0x1	Break enable bit: after software sets a breakpoint, the counter is prohibited from counting: 0: Counter operates normally. 1: Counter stops counting.
9:8	CKD	RW	0x0	Digital filter clock frequency division register (division ratio relative to CK_INT): 00: $t_{DTS} = t_{CK\_INT}$ 01: $t_{DTS} = 2 * t_{CK\_INT}$ 10: $t_{DTS} = 4 * t_{CK\_INT}$ 11: Reserved for future use (RFU), disabled
7	ARPE	RW	0x0	Auto-reload preload enable: 0: ARR register does not enable preload. 1: ARR register enables preload.
6:5	CMS	RW	0x0	Counter alignment mode selection: 00: Edge-aligned mode 01: Center-aligned mode 1, where the output compare interrupt flag is set only during down-counting. 10: Center-aligned mode 2, where the output compare interrupt flag is set only during up-counting. 11: Center-aligned mode 3, where the output compare interrupt flag is set during both up-counting and down-counting.
4	DIR	RW	0x0	Counting direction register: 0: Counting up 1: Counting down Note: This register is read-only when the timer is configured in center-aligned mode or encoder mode.

Bit	Name	Attribute	Reset Value	Description
3	OPM	RW	0x0	Single-pulse output mode: 0: The counter does not stop at the occurrence of an update event. 1: The counter stops (CEN automatically cleared) at the occurrence of an update event.
2	URS	RW	0x0	Update request selection: 0: An update interrupt or DMA request can be generated by any of the following events: ● Counter overflow or underflow ● Software setting the UG bit ● Slave controller generating an update 1: An update interrupt or DMA request can be generated only at counter overflow or underflow.
1	UDIS	RW	0x0	Update disable: 0: Enable update event; an update event can be generated by any of the following events: ● Counter overflow or underflow ● Software setting the UG bit ● Slave controller generating an update 1: Disable update event, do not update shadow register. The counter and prescaler will be reinitialized if the UG bit is set or the slave controller receives a hardware reset.
0	CEN	RW	0x0	Counter enable: 0: Counter is disabled. 1: Counter is enabled. Note: External trigger mode can automatically set CEN.

13.5.3 Control Register 2 GTIMER_CR2 (Offset: 04H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11	BPS4	RW	0x0	TI4 edge detection channel selection:

Bit	Name	Attribute	Reset Value	Description
				0: TRC channel 1: TRC mapped to TI4 edge detection (TI4F_ED)
10	BPS3	RW	0x0	TI3 edge detection channel selection: 0: TRC channel 1: TRC mapped to TI3 edge detection (TI3F_ED)
9	BPS2	RW	0x0	TI2 edge detection channel selection: 0: TRC channel 1: TRC mapped to TI2 edge detection (TI2F_ED)
8	BPS1	RW	0x0	TI1 edge detection channel selection: 0: TRC channel 1: TRC mapped to TI1 edge detection (TI1F_ED)
7	TI1S	RW	0x0	GTIMER input TI1 selection: 0: The GTIMER_CH1 pin is connected to TI1 input. 1: The GTIMER_CH1, CH2 and CH3 pins are connected to TI1 input via XOR.
6:4	MMS	RW	0x0	Master mode selection for configuring the source of the synchronization trigger signal (TRGO) sent to slave in master mode: 000: The UG bit of GTIMER_EGR is used as TRGO. 001: The counter enable signal CNE_EN is used as TRGO, which can be used to start multiple timers simultaneously. 010: The UE (update event) signal is used as TRGO. 011: Compare pulse, if the CC1IF flag is about to set, TRGO outputs a positive pulse. 100: OC1REF is used as TRGO. 101: OC2REF is used as TRGO. 110: OC3REF is used as TRGO. 111: OC4REF is used as TRGO. Note: The slave timer or ADC must have its working clock enabled in advance to receive the TRGO sent by the master timer.
3	CCDS	RW	0x0	Capture/compare DMA selection: 0: Send DMA request when a capture/compare event occurs. 1: Send DMA request when an update event occurs.

Bit	Name	Attribute	Reset Value	Description
2:0	RSV	-	-	Reserved

13.5.4 Slave Mode Control Register GTIMER_SMCR (Offset: 08H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	ETP	RW	0x0	External trigger signal polarity configuration: 0: Active on high level or rising edge 1: Active on low level or falling edge
14	ECE	RW	0x0	Enable clock enable: 0: Disable external clock mode 2 1: Enable external clock mode 2, with the counter clocked by any active edge of ETRF
13:12	ETPS	RW	0x0	External trigger signal prescaler register: The frequency of external trigger signal ETRP can be at most 1/4 of the GTIMER working clock frequency. When the input signal frequency is high, prescaling can be used. 00: No prescaler 01: Divided by 2 10: Divided by 4 11: Divided by 8
11:8	ETF	RW	0x0	External trigger signal filter frequency and length selection: 0000: No filter 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8

Bit	Name	Attribute	Reset Value	Description
				1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
7	MSM	RW	0x0	Master/slave mode selection: 0: No action, to perfectly synchronize the master timer and slave timer via TRGO (when perfectly synchronized, the master register CR2.MMM must be set to 010). 1: Delay the master TRGI trigger signal. Note: This is only applicable to the trigger mode in slave mode (SMS = 110) and is configured by the slave mode.
6:4	TS	RW	0x0	Trigger selection, used to select the trigger source for synchronizing the counter: 000: Internal trigger 0 (ITR0) 001: Internal trigger 0 (ITR1) 010: Internal trigger 0 (ITR2) 011: Internal trigger 0 (ITR3) 100: TI1 edge detection (TI1F_ED) 101: Filtered TI1 (TI1FP1) 110: Filtered TI2 (TI2FP2) 111: External trigger input (ETRF) Note: The TS bit can only be rewritten when the slave mode is disabled (i.e. SMS = 000).
3	RSV	-	-	Reserved
2:0	SMS	RW	0x0	Slave mode selection: 000: Slave mode disabled: after CEN is enabled, the prescaler is clocked directly by the internal clock. 001: Encoder mode 1: the counter counts up/down on the TI2FP2 edge depending on the TI1FP1 level. 010: Encoder mode 2: the counter counts up/down on the TI1FP1 edge depending on the TI2FP2 level. 011: Encoder mode 3: the counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the levels of other input signals.

Bit	Name	Attribute	Reset Value	Description
				100: Reset mode: the rising edge of TRGI initializes the counter and triggers a register update. 101: Gated mode: the counting clock is enabled when TRGI is high, and stopped when TRGI is low. 110: Trigger mode: the counter starts counting at the rising edge of TRGI (the counter does not reset). 111: External clock mode 1: the rising edge of TRGI directly drives the counter.

13.5.5 DMA and Interrupt Enable Register GTIMER_DIER (Offset: 0CH)

Bit	Name	Attribute	Reset Value	Description
31:15	RSV	-	-	Reserved
19	CC4BUEN	RW	0x1	DMA mode configuration for capture/compare channel 4: 0: Single transfer, DMA reads the value of the CCR4 register and transfers it to RAM. 1: Burst mode from RAM to GTIMER, with access address and length configured via DCR.
18	CC3BUEN	RW	0x1	DMA mode configuration for capture/compare channel 3: 0: CCR3 to RAM, DMA reads the value of the CCR3 register and transfers it to RAM. 1: Burst mode from RAM to GTIMER, with access address and length configured via DCR.
17	CC2BUEN	RW	0x1	DMA mode configuration for capture/compare channel 2: 0: CCR2 to RAM, DMA reads the value of the CCR2 register and transfers it to RAM. 1: Burst mode from RAM to GTIMER, with access address and length configured via DCR.

Bit	Name	Attribute	Reset Value	Description
16	CC1BUEN	RW	0x1	DMA mode configuration for capture/compare channel 1: 0: CCR1 to RAM, DMA reads the value of the CCR1 register and transfers it to RAM. 1: Burst mode from RAM to GTIMER, with access address and length configured via DCR.
15	RSV	-	-	Reserved
14	TDE	RW	0x0	External trigger DMA request enable: 0: In slave mode, disable DMA requests generated by external trigger events. 1: In slave mode, enable DMA requests generated by external trigger events (can be used to automatically update the preload register).
13	RSV	-	-	Reserved
12	CC4DE	RW	0x0	Capture/compare channel 4 DMA request enable: 0: Disable CC4 DMA request. 1: Enable CC4 DMA request.
11	CC3DE	RW	0x0	Capture/compare channel 3 DMA request enable: 0: Disable CC3 DMA request. 1: Enable CC3 DMA request.
10	CC2DE	RW	0x0	Capture/compare channel 2 DMA request enable: 0: Disable CC2 DMA request. 1: Enable CC2 DMA request.
9	CC1DE	RW	0x0	Capture/compare channel 1 DMA request enable: 0: Disable CC1 DMA request. 1: Enable CC1 DMA request.
8	UDE	RW	0x0	Update DMA request enable: 0: Disable DMA request generation on update event occurrence. 1: Enable DMA request generation on update event occurrence.
7	RSV	-	-	Reserved
6	TIE	RW	0x0	Trigger event interrupt enable: 0: Disable trigger event interrupt. 1: Enable trigger event interrupt.

Bit	Name	Attribute	Reset Value	Description
5	RSV	-	-	Reserved
4	CC4IE	RW	0x0	Capture/compare channel 4 interrupt enable: 0: Disable capture/compare channel 4 interrupt. 1: Enable capture/compare channel 4 interrupt.
3	CC3IE	RW	0x0	Capture/compare channel 3 interrupt enable: 0: Disable capture/compare channel 3 interrupt. 1: Enable capture/compare channel 3 interrupt.
2	CC2IE	RW	0x0	Capture/compare channel 2 interrupt enable: 0: Disable capture/compare channel 2 interrupt. 1: Enable capture/compare channel 2 interrupt.
1	CC1IE	RW	0x0	Capture/compare channel 1 interrupt enable: 0: Disable capture/compare channel 1 interrupt. 1: Enable capture/compare channel 1 interrupt.
0	UIE	RW	0x0	Update event interrupt enable: 0: Disable update event interrupt. 1: Enable update event interrupt.

13.5.6 Status Register GTIMER_SR (Offset: 10H)

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12	CC4OF	W1C	0x0	Capture/compare channel 4 overcapture status: Refer to CC1OF.
11	CC3OF	W1C	0x0	Capture/compare channel 3 overcapture status: Refer to CC1OF.
10	CC2OF	W1C	0x0	Capture/compare channel 2 overcapture status: Refer to CC1OF.
9	CC1OF	W1C	0x0	Capture/compare channel 1 overcapture status: This register is only valid when the corresponding channel is configured in input capture mode. This bit is set by hardware and cleared by software writing 1. 0: No overcapture event 1: A new capture occurs while the CC1IF flag is 1.
8:7	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
6	TIF	W1C	0x0	Trigger interrupt flag, set by hardware and cleared by software writing 1.
5	RSV	-	-	Reserved
4	CC4IF	W1C	0x0	Capture/compare channel 4 interrupt flag: Refer to CC1IF.
3	CC3IF	W1C	0x0	Capture/compare channel 3 interrupt flag: Refer to CC1IF.
2	CC2IF	W1C	0x0	Capture/compare channel 2 interrupt flag: Refer to CC1IF.
1	CC1IF	W1C	0x0	Capture/compare channel 1 interrupt flag: If CC1 channel is configured for output: CC1IF is set when the counter value equals the compare value, and cleared by software writing 1. If channel CC1 is configured as input: this flag is set by hardware on a capture. It is cleared by software via writing it to 1 or automatically cleared by software reading CCR1.
0	UIF	W1C	0x0	Update event interrupt flag, set by hardware and cleared by software writing 1. UIF is set and the shadow register is updated at the following events: Counter overflow occurs if repetition counter = 0 and UDIS = 0. The counter is initialized by software setting the UG bit if URS = 0 and UDIS = 0. The counter is initialized by a trigger event if URS = 0 and UDIS = 0.

13.5.7 Event Generation Register GTIMER_EGR (Offset: 14H)

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	TG	W	0x0	Software trigger: This bit is set by software to generate a trigger event, and automatically cleared by hardware.

Bit	Name	Attribute	Reset Value	Description
5	RSV	-	-	Reserved
4	CC4G	W	0x0	Capture/compare channel 4 software trigger, refer to CC1G
3	CC3G	W	0x0	Capture/compare channel 3 software trigger, refer to CC1G
2	CC2G	W	0x0	Capture/compare channel 2 software trigger, refer to CC1G
1	CC1G	W	0x0	Capture/compare channel 1 software trigger, automatically cleared by hardware: If CC1 channel is configured for output: CC1IF is set, and if enabled, it can generate the corresponding interrupt and DMA request. If CC1 channel is configured for input: the current counter value is captured into the CCR1 register, CC1IF is set, and if enabled, it can generate the corresponding interrupt and DMA request.
0	UG	W	0x0	Software update event: This bit can be set by software to generate an update event, and is automatically cleared by hardware. When the software sets UG, the counter is reinitialized, the shadow register is updated, and the prescaler counter is cleared.

13.5.8 Capture/Compare Mode Register 1 GTIMER_CCMR1 (Offset: 18H)

This register is multiplexed for two different functions under output compare and input capture modes:

1. Output compare mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
15	OC2CE	RW	0x0	Output compare 2 clear enable, refer to OC1CE
14:12	OC2M	RW	0x0	Output compare 2 mode configuration, refer to OC1M
11	OC2PE	RW	0x0	Output compare 2 preload enable, refer to OC1PE
10	OC2FE	RW	0x0	Output compare 2 fast enable, refer to OC1FE
9:8	CC2S	RW	0x0	Capture/compare channel 2 selection: 00: CC2 channel is configured as output. 01: CC2 channel is configured as input, IC2 is mapped on TI2. 10: CC2 channel is configured as input, IC2 is mapped on TI1. 11: CC2 channel is configured as input, IC2 is mapped on TRC (mapped on TI2 edge detection if BPS2 is set to 1). Note: CC2S can only be written when the channel is disabled (CC2E = 0).
7	OC1CE	RW	0x0	Output compare 1 clear enable: 0: OC1REF is not affected by ETRF input. 1: OC1REF is automatically cleared when ETRF is high.
6:4	OC1M	RW	0x0	Output compare 1 mode configuration: these bits define the behavior of the OC1REF signal. 000: The comparison result between the output compare register CCR1 and the counter CNT does not affect the output. 001: OC1REF is set high when CCR1 = CNT. 010: OC1REF is set low when CCR1 = CNT. 011: OC1REF is toggled when CCR1 = CNT. 100: OC1REF is fixed low (inactive). 101: OC1REF is fixed high (active). 110: PWM mode 1—In up-counting, OC1REF is set high when CNT < CCR1, otherwise set low; in down-counting, OC1REF is set low when CNT > CCR1, otherwise set high. 111: PWM mode 2—In up-counting, OC1REF is set low when CNT < CCR1, otherwise set high; in down-counting, OC1REF is set high when CNT > CCR1, otherwise set low.

Bit	Name	Attribute	Reset Value	Description
3	OC1PE	RW	0x0	Output compare 1 preload enable: 0: The CCR1 preload register is disabled, and CCR1 can be written directly. 1: The CCR1 preload register is enabled, and read/write operations on CCR1 access the preload register, with content shifted to the shadow register only when an update event occurs.
2	OC1FE	RW	0x0	Output compare 1 fast enable: 0: Disable fast enable, the trigger input will not affect the comparison output. 1: Enable fast enable, the trigger input will immediately change OC3REF to the output corresponding to the compare match, regardless of the actual current comparison status. This function acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC1S	RW	0x0	Capture/compare channel 1 selection: 00: CC1 channel is configured as output. 01: CC1 channel is configured as input, IC1 is mapped on TI1. 10: CC1 channel is configured as input, IC1 is mapped on TI2. 11: CC1 channel is configured as input, IC1 is mapped on TRC (if BPS1 is set to 1, then mapped on TI1 edge detection). Note: CC1S can only be written when the channel is disabled (CC1E = 0).

2. Input Capture Mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	IC2F	RW	0x0	Input capture 2 filter
11:10	IC2PSC	RW	0x0	Input capture 2 prescaler
9:8	CC2S	RW	0x0	Capture/compare channel 2 selection:

Bit	Name	Attribute	Reset Value	Description
				00: CC2 channel is configured as output. 01: CC2 channel is configured as input, IC2 is mapped on TI2. 10: CC2 channel is configured as input, IC2 is mapped on TI1. 11: CC2 channel is configured as input, IC2 is mapped on TRC (if BPS2 is set to 1, then mapped on TI2 edge detection). Note: CC2S can only be written when the channel is disabled (CC2E = 0).
7:4	IC1F	RW	0x0	Input capture 1 filter: This bit-field defines the sampling frequency and filter length for TI1. 0000: No filter, sampling at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
3:2	IC1PSC	RW	0x0	Input capture 1 prescaler: 00: No prescaling 01: Capture occurs once every 2 event inputs. 10: Capture occurs once every 4 event inputs. 11: Capture occurs once every 8 event inputs. Note: The IC1PSC register is reset when CC1E = 0.
1:0	CC1S	RW	0x0	Capture/compare channel 1 selection:

Bit	Name	Attribute	Reset Value	Description
				00: CC1 channel is configured as output. 01: CC1 channel is configured as input, IC1 is mapped on TI1. 10: CC1 channel is configured as input, IC1 is mapped on TI2. 11: CC1 channel is configured as input, IC1 is mapped on TRC (mapped on TI1 edge detection if BPS1 is set to 1). Note: CC1S can only be written when the channel is disabled (CC1E = 0).

13.5.9 Capture/Compare Mode Register 2 GTIMER_CCMR2 (Offset: 1CH)

This register is multiplexed for two different functions under output compare and input capture modes:

1. Output compare mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	OC4CE	RW	0x0	Output compare 4 clear enable, refer to OC1CE
14:12	OC4M	RW	0x0	Output compare 4 mode configuration, refer to OC1M
11	OC4PE	RW	0x0	Output compare 4 preload enable, refer to OC1PE
10	OC4FE	RW	0x0	Output compare 4 fast enable, refer to OC1FE
9:8	CC4S	RW	0x0	Capture/compare channel 4 selection: 00: CC4 channel is configured as output. 01: CC4 channel is configured as input, IC4 is mapped on TI4. 10: CC4 channel is configured as input, IC4 is mapped on TI3. 11: CC4 channel is configured as input, IC4 is mapped on TRC (mapped on TI4 edge detection if BPS4 is set to

Bit	Name	Attribute	Reset Value	Description
				1). Note: CC4S can only be written when the channel is disabled (CC4E = 0).
7	OC3CE	RW	0x0	Output compare 4 clear enable: 0: OC4REF is not affected by ETRF. 1: OC4REF is automatically cleared when ETRF is high.
6:4	OC3M	RW	0x0	Output compare 3 mode: these bits define the behavior of the OC3REF signal. 000: The comparison result between the output compare register CCR3 and the counter CNT does not affect the output. 001: OC3REF is set high when CCR3 = CNT. 010: OC3REF is set low when CCR3 = CNT. 011: OC3REF is toggled when CCR3 = CNT. 100: OC3REF is fixed low (inactive). 101: OC3REF is fixed high (active). 110: PWM mode 1—In up-counting, OC3REF is set high when CNT < CCR3, otherwise set low; in down-counting, OC3REF is set low when CNT > CCR3, otherwise set high. 111: PWM mode 2—In up-counting, OC3REF is set low when CNT < CCR3, otherwise set high; in down-counting, OC3REF is set high when CNT > CCR3, otherwise set low.
3	OC3PE	RW	0x0	Output compare 3 preload enable: 0: The CCR3 preload register is disabled; and CCR3 can be written directly. 1: The CCR3 preload register is enabled, and read/write operations on CCR3 access the preload register, with content shifted to the shadow register only when an update event occurs.
2	OC3FE	RW	0x0	Output compare 3 fast enable: 0: Disable fast enable, the trigger input will not affect the comparison output. 1: Enable fast enable, the trigger input will immediately change OC3REF to the output corresponding to the

Bit	Name	Attribute	Reset Value	Description
				compare match, regardless of the actual current comparison status. Note: This function acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC3S	RW	0x0	Capture/compare channel 3 selection: 00: CC3 channel is configured as output. 01: CC3 channel is configured as input, IC3 is mapped on TI3. 10: CC3 channel is configured as input, IC3 is mapped on TI4. 11: CC3 channel is configured as input, IC3 is mapped on TRC (mapped on TI3 edge detection if BPS3 is set to 1). Note: CC3S can only be written when the channel is disabled (CC3E = 0).

2. Input capture mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	IC4F	RW	0x0	Input capture 4 filter
11:10	IC4PSC	RW	0x0	Input capture 4 prescaler
9:8	CC4S	RW	0x0	Capture/compare channel 4 selection: 00: CC4 channel is configured as output. 01: CC4 channel is configured as input, IC4 is mapped on TI4. 10: CC4 channel is configured as input, IC4 is mapped on TI3. 11: CC4 channel is configured as input, IC4 is mapped on TRC (mapped on TI4 edge detection if BPS4 is set to 1). Note: CC4S can only be written when the channel is disabled (CC4E = 0).

Bit	Name	Attribute	Reset Value	Description
7:4	IC3F	RW	0x0	Input capture 3 filter: This bit-field defines the sampling frequency and filter length for TI3. 0000: No filter, sampling at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
3:2	IC3PSC	RW	0x0	Input capture 3 prescaler: 00: No prescaling 01: Capture occurs once every 2 event inputs. 10: Capture occurs once every 4 event inputs. 11: Capture occurs once every 8 event inputs. Note: The IC3PSC register is reset when CC3E = 0.
1:0	CC3S	RW	0x0	Capture/compare channel 3 selection: 00: CC3 channel is configured as output. 01: CC3 channel is configured as input, IC3 is mapped on TI3. 10: CC3 channel is configured as input, IC3 is mapped on TI4. 11: CC3 channel is configured as input, IC3 is mapped on TRC (mapped on TI3 edge detection if BPS3 is set to 1). Note: CC3S can only be written when the channel is disabled (CC3E = 0).

13.5.10 Capture/Compare Enable Register GTIMER_CCER (Offset: 20H)

Bit	Name	Attribute	Reset Value	Description
31:14	RSV	-	-	Reserved
13	CC4P	RW	0x0	Capture/compare 4 output polarity: refer to CC1P
12	CC4E	RW	0x0	Capture/compare 4 output enable: refer to CC1E
11:10	RSV	-	-	Reserved
9	CC3P	RW	0x0	Capture/compare 3 output polarity: refer to CC1P
8	CC3E	RW	0x0	Capture/compare 3 output enable: refer to CC1E
7:6	RSV	-	-	Reserved
5	CC2P	RW	0x0	Capture/compare 2 output polarity: refer to CC1P
4	CC2E	RW	0x0	Capture/compare 2 output enable: refer to CC1E
3:2	RSV	-	-	Reserved
1	CC1P	RW	0x0	Capture/compare 1 output polarity: When CC1 channel is configured as output: 0: OC1 is active high. 1: OC1 is active low. When CC1 channel is configured as input: 0: Non-inverted mode: capture occurs on the rising edge of IC1. 1: Inverted mode: capture occurs on the falling edge of IC1.
0	CC1E	RW	0x0	Capture/compare 1 output enable: When CC1 channel is configured as output: 0: OC1 is not active. 1: OC1 is active. When CC1 channel is configured as input: 0: Capture function is disabled. 1: Capture function is enabled.

13.5.11 Counter Register GTIMER_CNT (Offset: 24H)

Bit	Name	Attribute	Reset Value	Description
31:0	CNT	RW	0x0	Counter value

13.5.12 Prescaler Register GTIMER_PSC (Offset: 28H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	PSC	RW	0x0	Counter clock (CK_CNT) prescaler value: $f_{CK\_CNT} = f_{CK\_PSC} / (PSC[15:0] + 1)$ This is a preload register whose contents are transferred into the shadow register at each update event. Note: The maximum supported PWM output frequency is 30 MHz, which shall be taken into account when configuring PSC and ARR.

13.5.13 Auto-reload Register GTIMER_ARR (Offset: 2CH)

Bit	Name	Attribute	Reset Value	Description
31:0	ARR	RW	0x0	Auto-reload value at counter overflow. This is a preload register whose contents are transferred into the shadow register at each update event.

13.5.14 Capture/Compare Register 1 GTIMER_CCR1 (Offset: 34H)

Bit	Name	Attribute	Reset Value	Description
31:0	CCR1	RW	0x0	Capture/compare channel 1 register **If channel CC1 is configured as output:** This is a preload register whose contents are loaded into the shadow register and then compared with the counter to generate the OC1 output. **If channel CC1 is configured as input:**

Bit	Name	Attribute	Reset Value	Description
				CCR1 stores the counter value at the time of the most recent input capture event, at which point CCR1 is read-only.

13.5.15 Capture/Compare Register 2 GTIMER_CCR2 (Offset: 38H)

Bit	Name	Attribute	Reset Value	Description
31:0	CCR2	RW	0x0	Capture/compare channel 2 register **If channel CC2 is configured as output:** This is a preload register whose contents are loaded into the shadow register and then compared with the counter to generate the OC2 output. **If channel CC2 is configured as input:** CCR2 stores the counter value at the time of the most recent input capture event, at which point CCR2 is read-only.

13.5.16 Capture/Compare Register 3 GTIMER_CCR3 (Offset: 3CH)

Bit	Name	Attribute	Reset Value	Description
31:0	CCR3	RW	0x0	Capture/compare channel 3 register **If channel CC3 is configured as output:** This is a preload register whose contents are loaded into the shadow register and then compared with the counter to generate the OC3 output. **If channel CC3 is configured as input:** CCR3 stores the counter value at the time of the most recent input capture event, at which point CCR3 is read-only.

13.5.17 Capture/Compare Register 4 GTIMER_CCR4 (Offset: 40H)

Bit	Name	Attribute	Reset Value	Description
31:0	CCR4	RW	0x0	Capture/compare channel 4 register

Bit	Name	Attribute	Reset Value	Description
				If channel CC4 is configured as output: This is a preload register whose contents are loaded into the shadow register and then compared with the counter to generate the OC4 output. **If channel CC4 is configured as input:** CCR4 stores the counter value at the time of the most recent input capture event, at which point CCR4 is read-only.

13.5.18 DMA Control Register GTIMER_DCR (Offset: 48H)

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12:8	DBL	RW	0x0	DMA burst length: Reading from or writing to the MAR register will trigger a burst DMA operation, with the burst length ranging from 1 to 18. 00000: Length = 1 00001: Length = 2 …… 10001: Length = 18 Others: Invalid values, writing is prohibited.
7:5	RSV	-	-	Reserved
4:0	DBA	RW	0x0	DMA base address, defining the offset address directed to the register: 00000: CR1 00001: CR2 00010: SMCR …… 01011: ARR 01100: RSV 01101: CCR1 …… 10000: CCR4

Bit	Name	Attribute	Reset Value	Description
				10001: RSV 11000: DCR Note: If DBA + DBL exceeds the address range of GTIMER register, the actual burst transfer will stop automatically after reaching the highest address of GTIMER register, i.e., the burst length will be shortened.

13.5.19 DMA Access Register GTIMER_DMAR (Offset: 4CH)

Bit	Name	Attribute	Reset Value	Description
31:0	DMAR	RW	0x400	DMA burst access register When using DMA burst transfer, set the DMA channel peripheral address to DMAR, and GTIMER will generate multiple DMA requests based on the value of DBL.

14 General-purpose Timers GTIMER1/2

14.1 Overview

The general-purpose timers/counters (GTIMER1/2) feature a 16-bit auto-reload counter and a programmable prescaler, with its own independent interrupt. It serves various purposes, including measuring the pulse width of input signals (input capture) and generating output waveforms. The counter can count up, down or up/down with the counter value accessible by software at any time.

14.2 Main Features

- 16-bit up, down, up/down auto-reload counter
- 16-bit programmable prescaler allowing real-time adjustment of the counter clock frequency division
- Flexible selection of counter clock source
- Channels available for input capture, output compare, PWM (edge- or center-aligned mode), and single-pulse output
- Supporting cascading between timers
- Interrupt generated on the following events:
  - Update: counter up/down overflow
  - Input capture
  - Output compare
- Supporting incremental quadrature encoder and Hall sensor

14.3 GTIMER1/2 Block Diagram

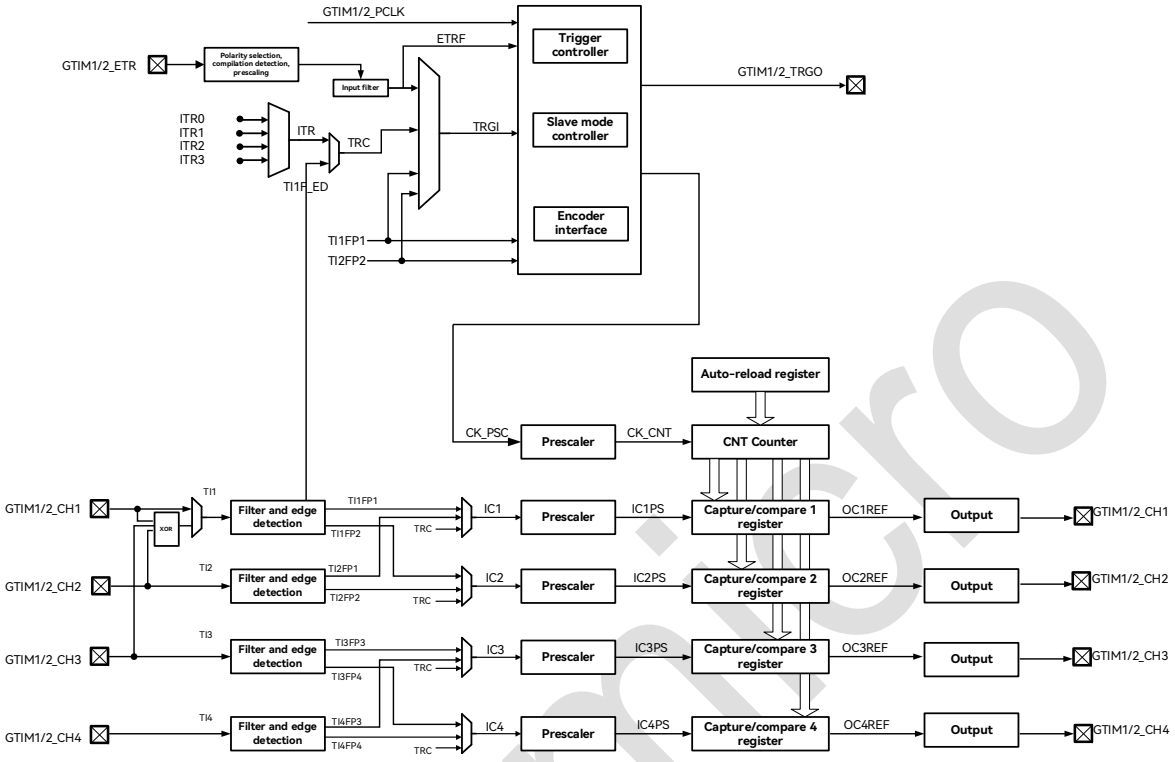


Figure 14-1: GTIMER Block Diagram

14.4 Functional Description

14.4.1 Timing Unit

The main components of GTIMER1/2 is a 16-bit counter and its associated auto-reload register. This counter can count up, down, or both up and down. Its clock is derived from a prescaler. The counter, auto-reload register and prescaler can be read and written by software even when the counter is running.

The timing unit consists of the following registers:

- Counter register (GTIMER_CNT)
- Prescaler register (GTIMER_PSC)

- Auto-reload register (GTIMER_ARR)

The GTIMER_ARR features a preload feature. According to the setting of the auto-reload preload enable bit (ARPE) in the GTIMER_CR1 register, the content of the preload register is transferred to the shadow register either immediately or upon each update event (UEV). An update event occurs when the counter reaches the overflow condition (underflow condition during down-counting) and when the UDIS bit in the GTIMER_CR1 register is 0. Software can also actively trigger an update of ARR by writing GTIMER_EGR[0] to 1.

The operating clock of the counter is driven by the divided clock generated by GTIMER_PSC. GTIMER_CNT starts counting only when the counter enable bit (CEN) is set.

GTIMER_PSC is a synchronous prescaler, which can divide the counter clock frequency by any value between 1 and 65536. It is a 16-bit counter controlled by a 16-bit register. Since this control register is equipped with a buffer, it can be modified on the fly. The new prescaler parameters take effect at the next update event.

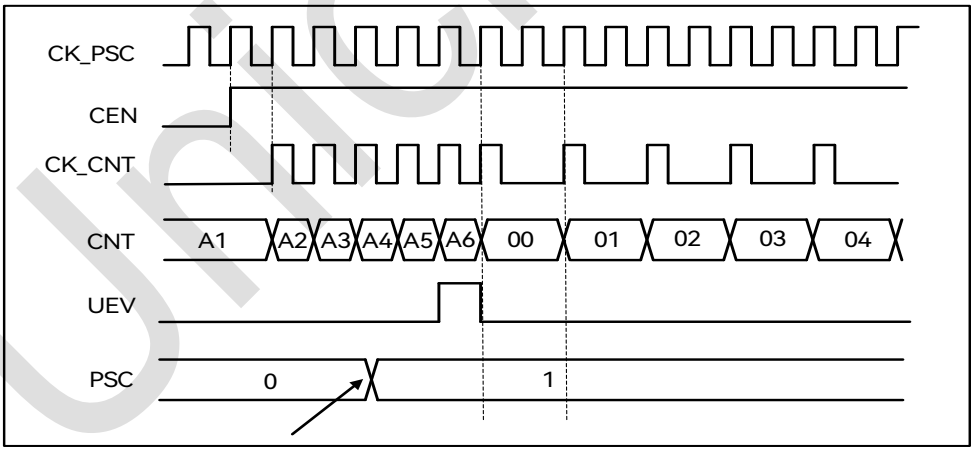


Figure 14-2: GTIMER Timing Diagram with Prescaler Division Changing from 1 to 2

14.4.2 Timer Operating Modes

The counter supports up-counting mode, down-counting mode and center-aligned mode.

14.4.2.1 Up-counting Mode

In up-counting mode, the counter counts from 0 to the auto-reload value (the value in the ARR register), then restarts counting from 0, generating a counter overflow event.

Setting the UG bit in the GTIMER_EGR register (either through software or using the slave mode controller can also generate an update event. Setting the UDIS bit in the GTIMER_CR1 register can disable the update event, preventing the shadow register from being updated when writing new values into the preload register. No update events will be generated until the UDIS bit is cleared. However, when an update event should occur, the counter will still be reset to 0, and the prescaler counter will also be reset to 0 (though the prescaler value remains unchanged).

If the URS bit in the GTIMER_CR1 register is set, setting the UG bit will generate an update event (UEV), but the hardware will not set the UIF flag.

When an update event occurs, the following registers are updated, and the UIF flag is set by hardware simultaneously:

- The shadow register of GTIMER_ARR is updated with the content of GTIMER_ARR.
- The shadow register of GTIMER_PSC is updated with the content of GTIMER_PSC.

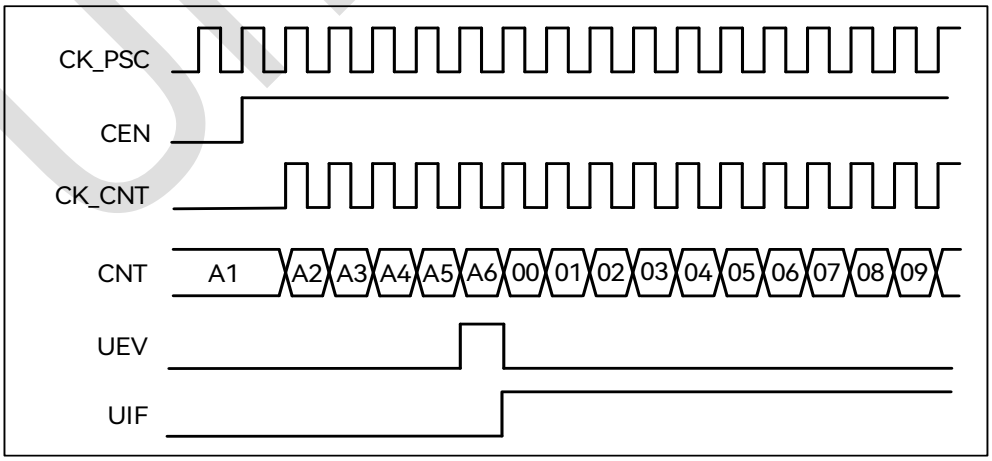


Figure 14-3: Up-counting Waveform

14.4.2.2 Down-counting Mode

In down-counting mode, the counter counts down from the value in GTIMER_ARR to 0, then restarts counting from the value in GTIMER_ARR, generating a counter underflow event.

Setting the UG bit in the GTIMER_EGR register (either through software or using the slave mode controller can also generate an update event. Setting the UDIS bit in the GTIMER_CR1 register can disable the update event, preventing the shadow register from being updated when writing new values into the preload register. No update events will be generated until the UDIS bit is cleared. However, when an update event should occur, the counter will still be reset to 0, and the prescaler counter will also be reset to 0 (though the prescaler value remains unchanged).

If the URS bit in the GTIMER_CR1 register is set, setting the UG bit will generate an update event (UEV), but the hardware will not set the UIF flag.

When an update event occurs, the following registers are updated, and the UIF flag is set by hardware simultaneously:

- The shadow register of GTIMER_ARR is updated with the content of GTIMER_ARR.
- The shadow register of GTIMER_PSC is updated with the content of GTIMER_PSC.

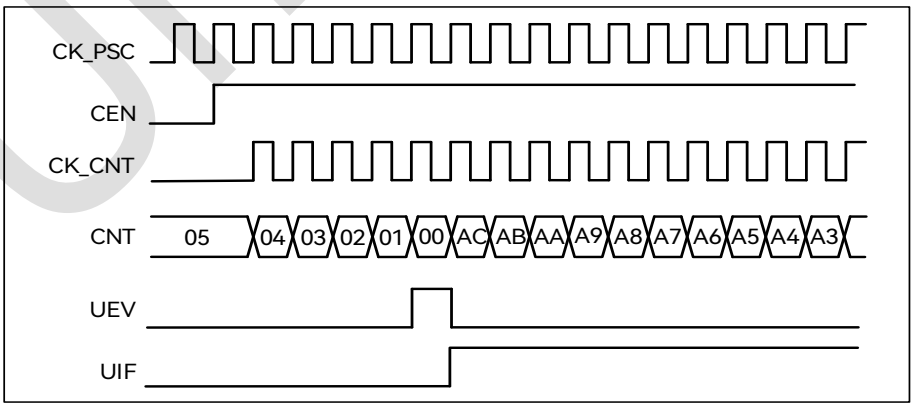


Figure 14-4: Down-counting Waveform

14.4.2.3 Center-aligned Counting Mode

In the center-aligned mode, the counter starts counting up from 0 to GTIMER_ARR-1, generating an overflow event, then starts counting down from GTIMER_ARR to 1, generating an underflow event, and then restarts counting up from 0.

The CR1[6:5] register is used to enable the center-aligned mode and select the output compare mode therein:

- CMS = 00: Edge-aligned mode
- 01: Center-aligned mode 1, where the output compare interrupt flag is set only during down-counting.
- 10: Center-aligned mode 2, where the output compare interrupt flag is set only during up-counting.
- 11: Center-aligned mode 3, where the output compare interrupt flag is set during both up-counting and down-counting.

In center-aligned mode, the DIR direction bit in the register cannot be rewritten by software, it is automatically updated by hardware as the counting direction changes, indicating the current counting direction. The following registers are updated on both overflow and underflow events of the counter:

- The shadow register of GTIMER_ARR is updated with the content of GTIMER_ARR.
- The shadow register of GTIMER_PSC is updated with the content of GTIMER_PSC.

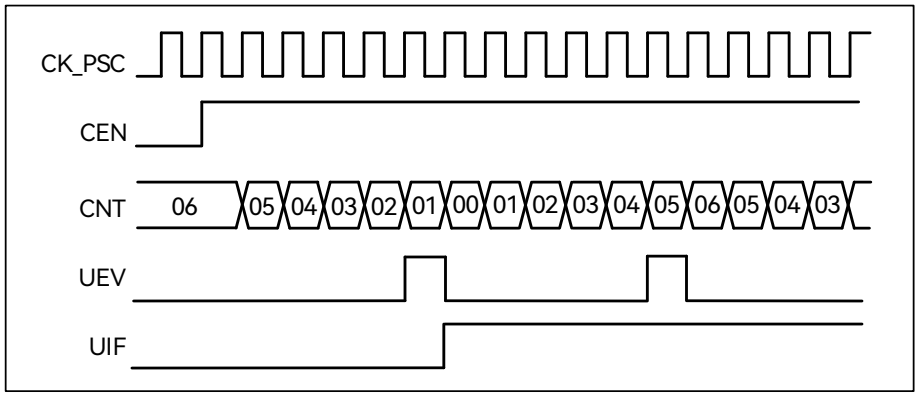


Figure 14-5: Center-aligned Counter Timing Diagram

14.4.3 Counter Clock

The counter can operate with the following clocks:

- Internal clock APBCLK (CK_INT)
- External pin input clock GTIMER_CHx (x = 1, 2)
- External pin trigger input (GTIMER_ETR)
- Internal trigger (ITRx)

14.4.3.1 APBCLK

When APBCLK is selected as the clock source, setting the GTIMER to slave mode (SMS = 000) is prohibited. Register bits such as CEN, DIR and UG are all controlled by software. After the software operates the UG register, the frequency divider and counter value will be reinitialized.

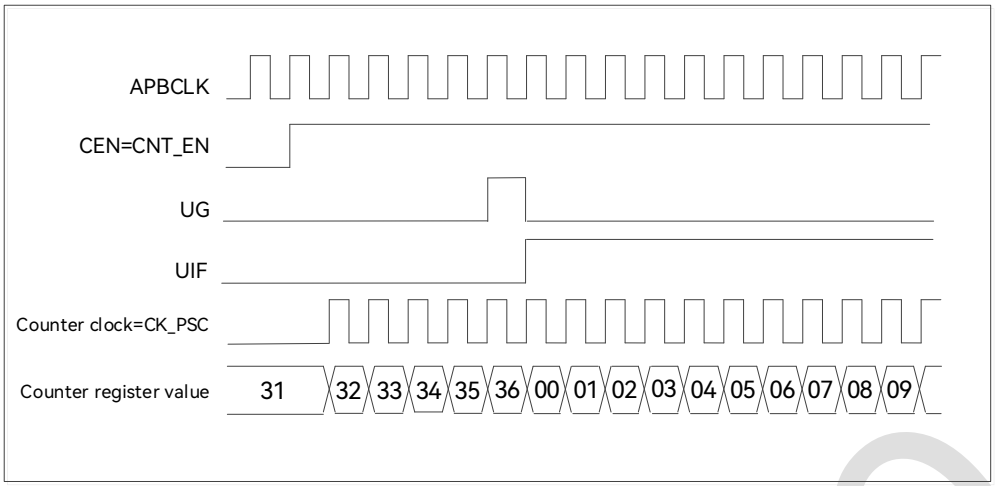


Figure 14-6: APBCLK Clock Source Mode with a Clock Prescaler Factor of 1

14.4.3.2 GTIMER_CHx (x = 1, 2) Pin Input: External Clock Mode 1

When configuring SMS = 111 in this mode, the input signal of the external pin GTIMER_CHx is directly used as the counting clock, with SMS configured to 111, and the counting edge can be selected as either the rising or falling edge.

In this mode, only the inputs of channels 1 and 2 can be used as clock inputs. Taking GTIMER_CH1 as an example, the configuration process is as follows:

1. Configure the corresponding pin as GTIMER_CH2 function in the IO module.
2. Configure GTIMER->CCMR1[9:8] = 01 to set channel 2 to detect the rising edge of TI2 input.
3. Configure GTIMER->CCMR1[15:12] to the desired filter value to select the input filter bandwidth (if no filter is required, keep CCMR1[15:12] = 0000).
4. Configure GTIMER->CCER[5] = 0 to set the rising edge polarity of TI2.
5. Configure GTIMER->SMCR[2:0] = 111 to select the timer external clock mode 1.
6. Configure GTIMER->SMCR[6:4] = 110 to select TI2 as the trigger input source.
7. Configure GTIMER-> CCER[0] = 1 to enable the channel.
8. Configure GTIMER-> CR1[0] = 1 to enable the counter.

14.4.3.3 GTIMER_ETR Pin Input: External Clock Mode 2

In this mode, the rising edge or falling edge of the input signal at the GTIMER_ETR pin is used for counting. The delay between the rising edge of ETR and the actual counter clock depends on the synchronization circuit at the ETRP signal terminal.

When using external clock mode 2, the GTIMER can still be configured in slave mode: for example, using the GTIMER_ETR input for counting while using the TRGO of another timer as a trigger signal. When a trigger event occurs, the counter is reset and starts counting again.

14.4.4 Internal Trigger Signal (ITRx)

The GTIMER supports four internal trigger inputs, which can be used for counting triggers or internal signal capture. When used for internal signal capture, TS must be configured to 000–011 for selecting ITR0–ITR3, and CCxS must be configured to 11 for selecting TRC as the capture signal.

Each ITR input supports four internal signal extensions, configured by the TS register. Please refer to the table below for input signal sources:

Table 14-1: GTIMER0/1/2 Internal Signal Extension Table

Slave	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
GTIMER0	RCL32k	ATIMER_TRGO	COMP0_IN	COMP1_IN
GTIMER1	RCL32k	GTIMER0_TRGO	COMP0_IN	COMP1_IN
GTIMER2	RCL32k	GTIMER1_TRGO	COMP0_IN	COMP1_IN

14.4.5 Capture/Compare Channels

GTIMER1 includes 4 capture/compare channels, each of which consists of a capture/compare register CCR (including a shadow register), an input circuit (digital filter, multiplexer, and prescaler), and an output circuit (comparator and output control).

14.4.5.1 Capture Channel

The input circuit samples the corresponding TIx (x = 1...4) and generates a filtered signal TIxF (x = 1...4), which is then processed to generate the corresponding TIxFPx (x = 1...4) signal through edge detection and polarity selection. This signal can be used as a counting trigger source or a capture source, and it enters the capture channel after passing through the prescaler.

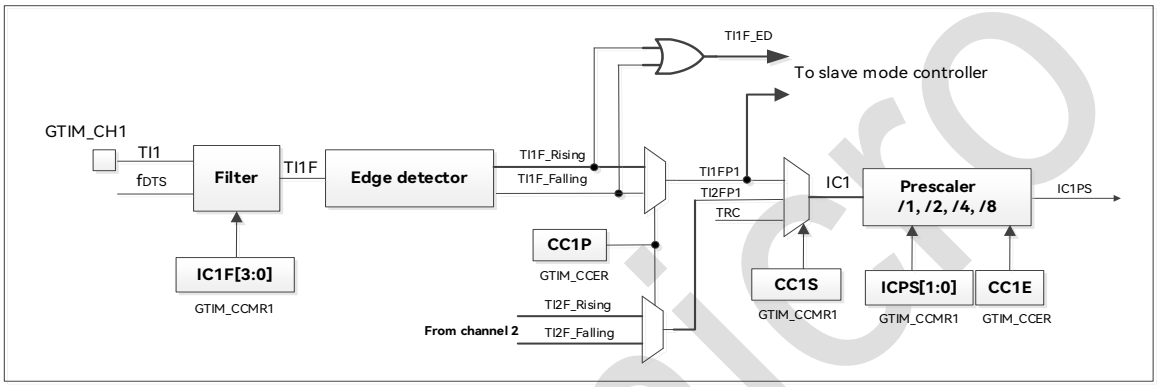


Figure 14-7: Capture/Compare Channel (Input Section of Channel 1)

14.4.5.2 Compare Channel

The output stage circuit generates an intermediate signal OCxREF (active high) as a reference, which serves as the reference input for the final output circuit.

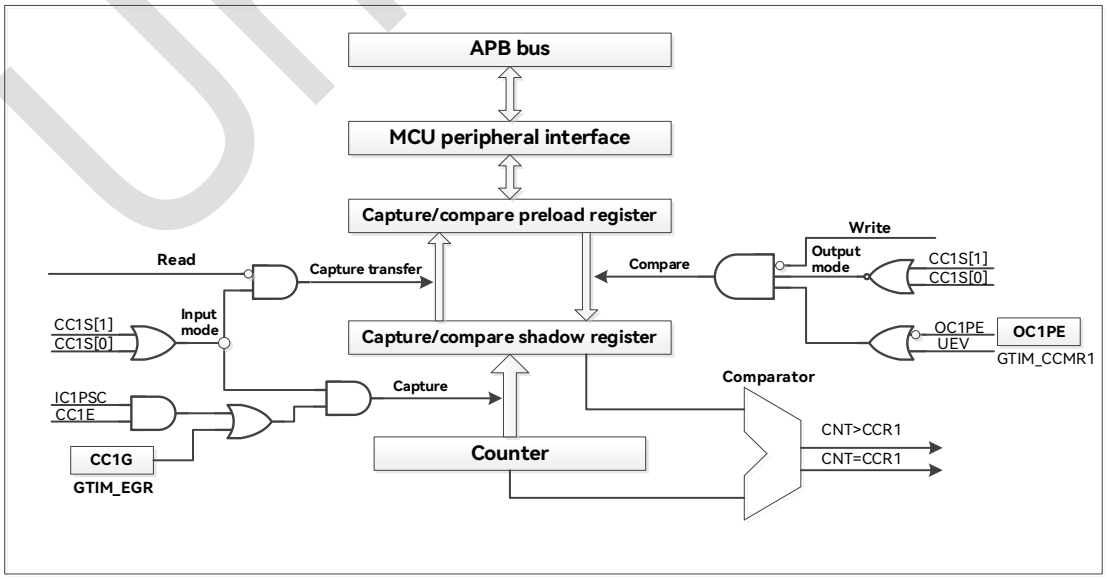


Figure 14-8: Main Circuit of Capture/Compare Channel 1

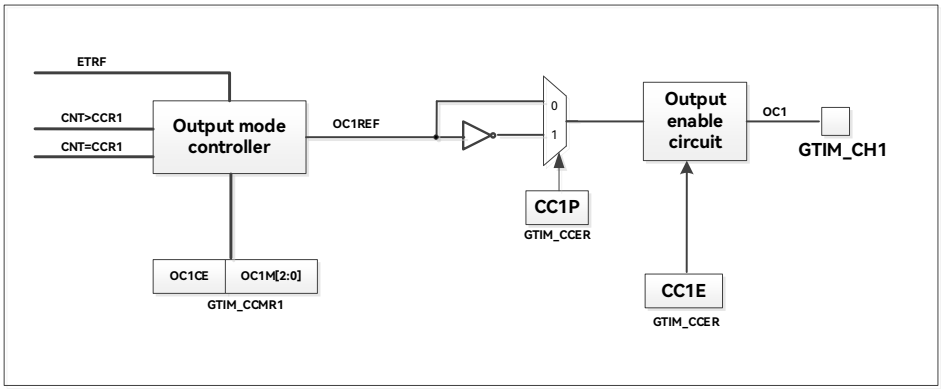


Figure 14-9: Output Section of Capture/Compare Channel

The capture/compare module consists of a preload register and a shadow register. Read and write operations only act on the preload register. In capture mode, capture occurs on the shadow register, and the captured value is then copied to the preload register. In compare mode, the content of the preload register is copied to the shadow register, and then the content of the shadow register is compared with the counter.

14.4.6 Input Capture Mode

In input capture mode, when the corresponding edge is detected on the ICx signal, the current value of the counter is latched into the capture/compare register (GTIMER_CCRx). When a capture event occurs, the corresponding CCxIF flag (in the GTIMER_SR register) is set to 1. If interrupts or DMA are enabled, an interrupt or DMA request will be generated. If the CCxIF flag is already high when a capture event occurs, the overcapture flag CCxOF (in the GTIMER_SR register) is set to 1. Writing to CCxIF or reading the capture data stored in the GTIMER_CCRx register can also clear CCxIF. Writing CCxOF = 1 can clear CCxOF.

The following example illustrates how to capture the counter value on the rising edge of TI1 input, with the steps as follows:

1. In GPIO module, configure the corresponding pin for GTIMER_CH1 function.
2. Select the input channel by setting GTIMER->CCMR1.CC1S = 01, mapping IC1 to TI1.

3. Select the active counting edge to be the rising edge for the T1 channel by setting GTIMER_CCER.CC1P = 0.
4. Configure the input filter time by setting GTIMER_CCMR1.IC1F.
5. Configure the input prescaler by setting GTIMER_CCMR1.IC1PS.
6. Set CC1E = 1 in the GTIMER_CCER register to enable capturing the counter value into the capture register.
7. Enable the corresponding interrupt or DMA request as required.
8. Enable the channel by setting GTIMER_CCER[0] = 1.

14.4.7 PWM Input Mode

This mode is a special case of input capture mode, where the pulse width of a PWM signal can be captured using two channels in cooperation.

To implement PWM input capture, the configuration process is as follows:

1. In GPIO module, configure the corresponding pin for GTIMER_CH1 function.
2. Select the valid input for GTIMER_CCR1 by setting CC1S = 01 (selecting TI1) in the GTIMER_CCMR1 register.
3. Select the valid polarity for TI1FP1 by setting CC1P = 0 (rising edge active).
4. Select the valid input for GTIMER_CCR2 by setting CC2S = 10 (selecting TI1) in the GTIMER_CCMR1 register.
5. Select the valid polarity for TI1FP2 by setting CC2P = 1 (falling edge active).
6. Select the valid trigger input signal by setting TS = 101 in the GTIMER_SMCR register.
7. Configure the slave mode controller to reset mode by setting SMS = 100 in the GTIMER_SMCR register.

8. Enable capture by setting CC1E = 1 and CC2E = 1 in the GTIMER_CCER register.

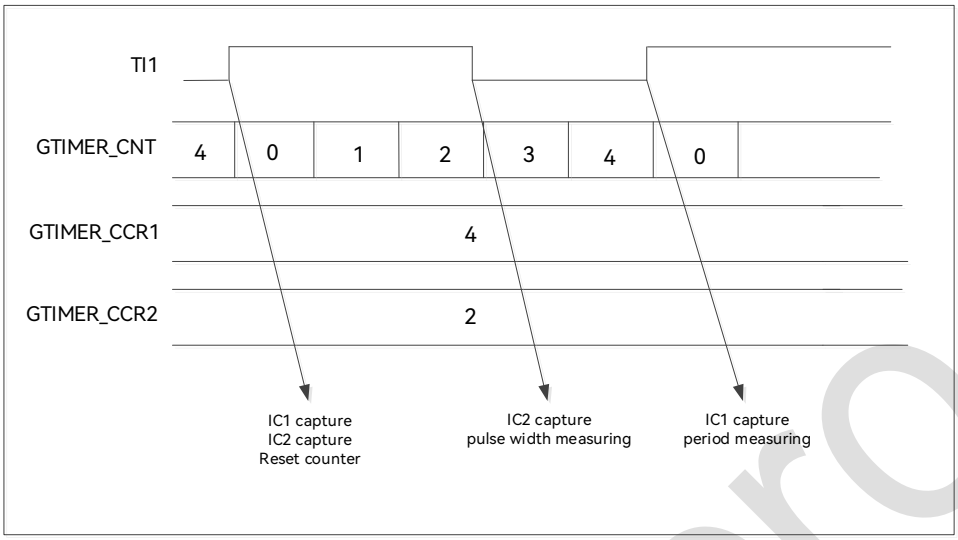


Figure 14-10: PWM Input Capture

14.4.8 Software-forced Output

In forced output mode, the output compare signals (OCxREF and the corresponding OCx/OCxN) can be directly forced to an active or inactive state by software, independent of the comparison result between the output compare register and the counter.

Setting the corresponding OCxM = 101 in the GTIMER_CCMRx register forces the output compare signal to an active state. In this case, OCxREF is forced to a high level, while OCx receives a signal with the opposite polarity of CCxP. For example: if CCxP = 0 (OCx active high), OCx is forced to a high level. Setting OCxM = 100 in the GTIMER_CCMRx register forces OCxREF to a low level.

14.4.9 Output Compare Mode

In output compare mode, when a match is found between CCR and the counter, OCxREF can be set to active, inactive or toggle on match. At the same time, the interrupt flag is also set and DMA requests can be sent. The output compare can also be used to output a pulse signal

of a specific width (in single-pulse mode).

Operation procedure:

1. Select the counting clock (internal, external, prescaler, etc.).
2. Set the values of GTIMER_ARR and GTIMER_CCR registers.
3. Enable corresponding interrupts and DMA request as required.
4. Select the output mode, for example:

- Set OCxM = 011 to toggle the OCx output pin when the counter matches CCRx.
- Set OCxPE = 0 to disable the preload register.
- Set CCxP = 0 to select the active-high polarity.
- Set CCxE = 1 to enable output.

Set CEN = 1 to enable the counter.

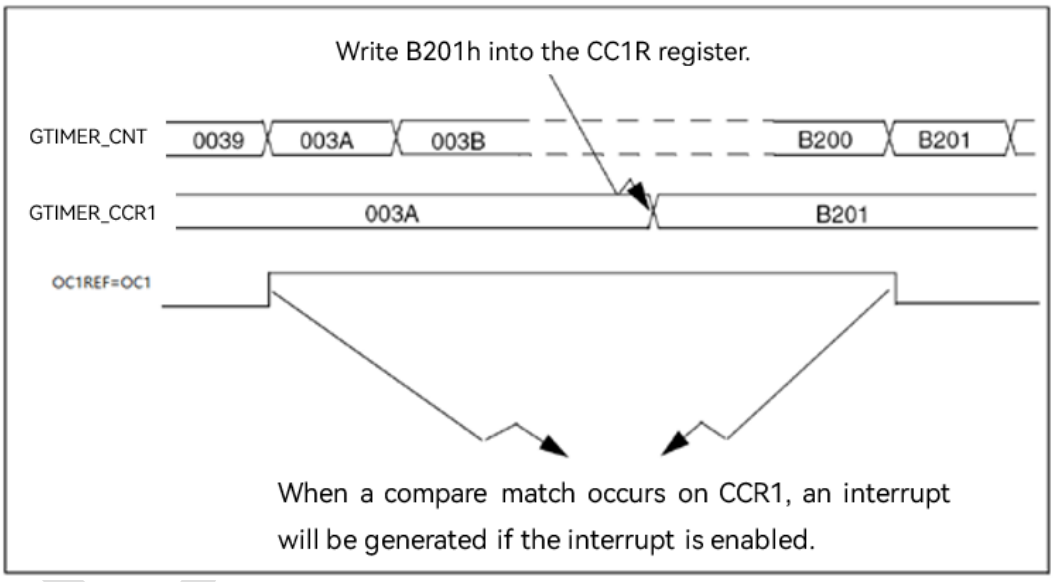


Figure 14-11: Output Compare Mode, Toggle on OC1

If the preload function is disabled, the CCR register can be rewritten by software at any time to achieve real-time control of the output waveform. Otherwise, the CCR shadow register is only updated with the content of the preload register at the next update event.

14.4.10 PWM Output

The PWM mode can generate pulse-width modulation signals with the period determined by the ARR register and the duty cycle determined by the CCR register. The polarity of the output signal can be configured via the CCxP bit in the register. In PWM mode, CNT and CCR registers are always compared. The counter supports edge-aligned and center-aligned counting modes; accordingly, the PWM output also supports these two aligned modes.

14.4.10.1 PWM Edge-aligned Mode

In up-counting mode, when it is configured in PWM mode 1, the OCxREF signal is high when CNT < CCR, otherwise it is low. And OCxREF will be fixed at 1 if CCR > ARR, and fixed at 0 if CCR is 0.

In down-counting mode, the definition of OCxREF level (high/low) is the same as that in up-counting mode.

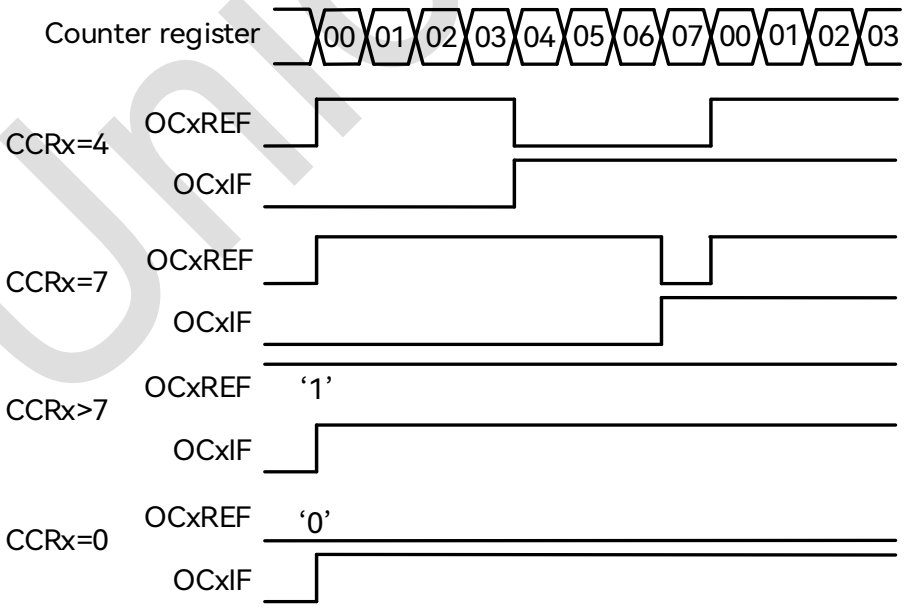


Figure 14-12: Edge-aligned PWM Waveform (ARR = 7)

14.4.10.2 PWM Center-aligned Mode

The definition of OCxREF level is the same as that in edge-aligned mode.

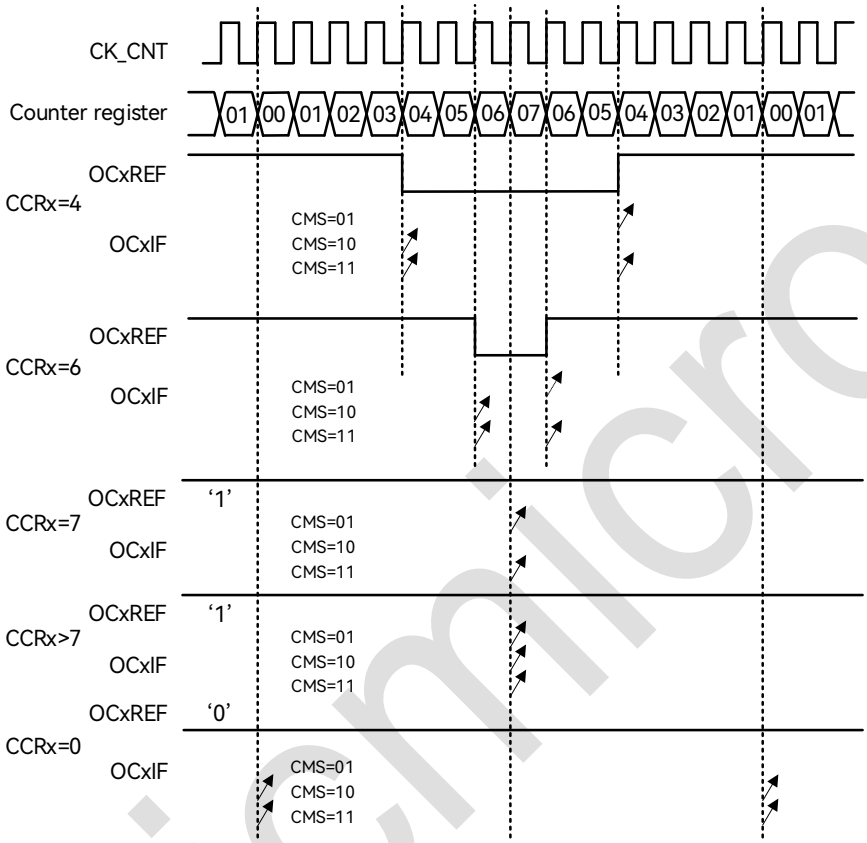


Figure 14-13: Center-aligned PWM Waveform (APR = 7)

When start counting in center-aligned mode, the initial counting direction is determined by the DIR bit in the register, and in the subsequent process, the DIR bit is directly controlled by hardware. The safest way to use center-aligned mode is to generate an update by setting the UG bit in the register just before starting the counter and not to overwrite the counter while it is running.

14.4.11 Output Compare Fast Enable Mode

When the fast enable mode is enabled (CCMR1[2,10], CCMR2[2,10]), the comparison output result of GTIMER is affected by the TRGI signal. The selection of TRGI signal is set through the SMCR[4:6] register. Taking channel 1 as an example:

Assuming CC1P is 0:

- After the TRGI signal arrives, if PWM mode 1 is adopted, regardless of the original level of PWM, GTIMER_CH1 is forced to a low level (0), and returns to normal once the count reaches ARR.
- After the TRGI signal arrives, if PWM mode 2 is adopted, regardless of the original level of PWM, GTIMER_CH1 is forced to a low level (1), and returns to normal once the count reaches ARR.

Assuming CC1P is 1:

- After the TRGI signal arrives, if PWM mode 1 is adopted, regardless of the original level of PWM, GTIMER_CH1 is forced to a low level (1), and returns to normal once the count reaches ARR.
- After the TRGI signal arrives, if PWM mode 2 is adopted, regardless of the original level of PWM, GTIMER_CH1 is forced to a low level (0), and returns to normal once the count reaches ARR.

14.4.12 Single-pulse Output Mode

Single-pulse output is a special case of the compare output mode, allowing users to output a pulse signal with a programmable width after a programmable delay following the occurrence of a specified event.

Unlike other output modes, the counter will automatically stop at the next update event. A pulse can be correctly generated only if the initial values of CCR and CNT are different. In up-counting, it is required that CNT < CCR ≤ ARR; in down-counting, it is required that CNT > CCR.

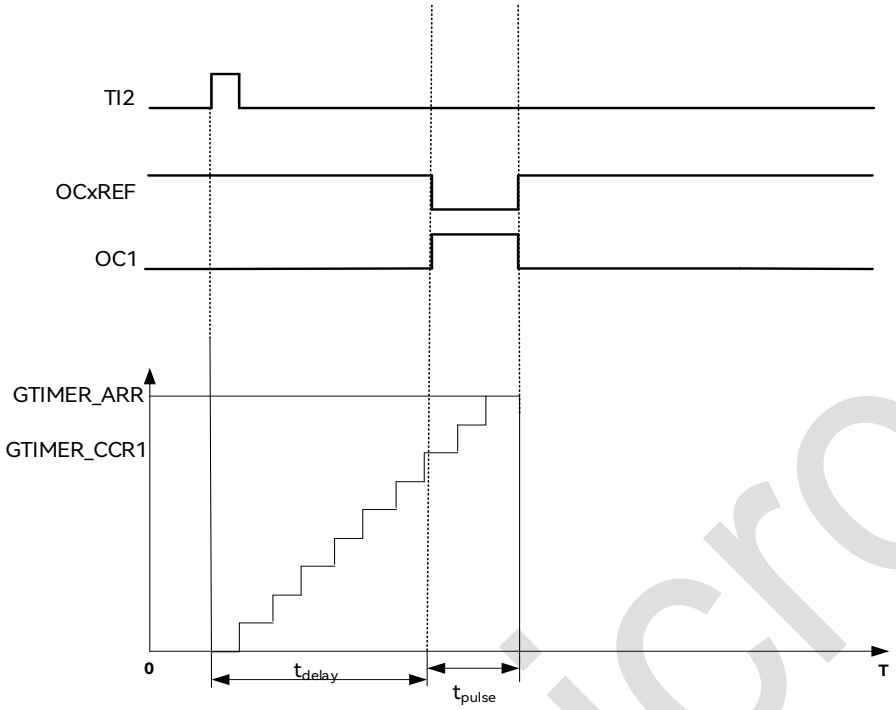


Figure 14-14: Timing Diagram of Single-pulse Mode

Figure 14-14 shows that TI2 input is used as the counter trigger signal. When the counter value equals CCR, OCxREF outputs a low level. When the counter reaches ARR, OCxREF returns to a high level, and the counter returns to 0 and stops counting.

Configuration for realizing the above functions with TI2 as input trigger:

- In GPIO module, configure the corresponding pin for GTIMER_CH2 function.
- Disable the channel by setting GTIMER->CCER.CC2E = 0 to ensure the success of subsequent configurations.
- Select the input channel by setting GTIMER->CCMR1.CC2S = 01.
- Select the active counting edge by setting GTIMER->CCER.CC2P = 0.
- Select TI2FP2 as the trigger input source by setting GTIMER->SMCR.TS[2:0] = 110.
- Set the slave mode controller to trigger mode by setting GTIMER->SMCR.SMS[2:0] = 110, with TI2FP2 for activating the counter.

- Enable the channel by setting GTIMER->CCER.CC2E = 1.

Configuration for realizing the above functions with OC1 as output:

- In GPIO module, configure the corresponding pin for GTIMER_CH1 function.
- Disable the channel by setting GTIMER->CCER.CC1E = 0 to ensure the success of subsequent configurations.
- Configure the output channel by setting GTIMER->CCMR1.CC1S = 00.
- Select the active counting edge by setting GTIMER->CCMR1.OC1M = 111, for PWM mode 2.
- Enable the channel by setting GTIMER->CCER.CC1E = 1.

Special settings for generating a single-pulse waveform base:

- t_{delay} is determined by the value of GTIMER->CCR1.
- t_{pulse} is determined by the difference between GTIMER->ARR and GTIMER->CCR1 (calculated as GTIMER->ARR - GTIMER->CCR1).
- Configure to single-pulse mode by setting GTIMER_CR1.OPM = 1.

14.4.13 Clearing OCxREF via External Event

OCxREF is active at high level, and it can be directly pulled down until the next update event by applying a high level to the external GTIMER_ETR pin. This function is only effective in output compare and PWM modes, and does not work in software-forced mode. To enable this function, OCxCE must be set to 1.

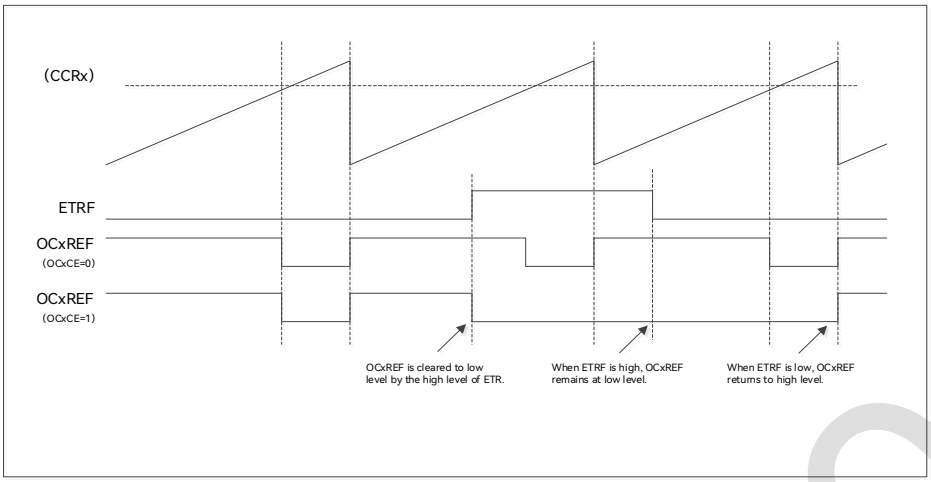


Figure 14-15: GTIMER_ETR Signal Clearing OCxREF of GTIMER

14.4.14 Encoder Interface Mode

The encoder interface mode involves two external input signals. GTIMER determines whether to increment or decrement the counter value based on the edge of one signal relative to the level of the other signal. The following table shows the relationship between the counting modes and the two input signals:

Table 14-2: Encoder Counting Methods and Input Signals

Counting Signal	TI2 Level When Counting on TI1/ TI1 Level When Counting on TI2	TI1 Signal		TI2 Signal	
		Rising Edge	Falling Edge	Rising Edge	Falling Edge
Counting on TI1	H	Decrement	Increment	No count	No count
	L	Increment	Decrement	No count	No count
Counting on TI2	H	No count	No count	Increment	Decrement
	L	No count	No count	Decrement	Increment
Counting on TI1 and TI2	H	Decrement	Increment	Increment	Decrement
	L	Increment	Decrement	Decrement	Increment

For example, when the counter is counting on TI1, it will count down if TI2 is sampled as high level on the rising edge of TI1, and count up if TI2 is sampled as high level on the falling edge of TI1.

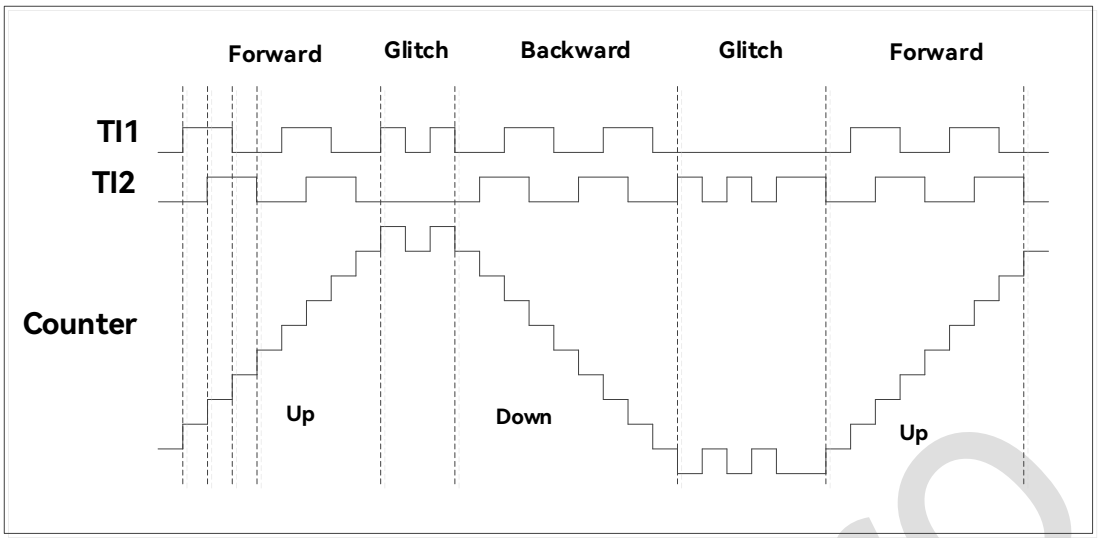


Figure 14-16: Example of Counter Operation in Encoder Interface Mode

The input channels in encoder interface mode shall be configured as follows:

- In GPIO module, configure the corresponding pins for GTIMER_CH1 and GTIMER_CH2 functions.
- Disable the channel by setting GTIMER->CCER.CC1E = 0 and GTIMER->CCER.CC2E = 0 to ensure the success of subsequent channel configurations.
- Select the input channel by setting GTIMER->CCMR1.CC1S = 01 and GTIMER->CCMR1.CC2S = 01.
- Select the active counting edge by setting GTIMER->CCER.CC1P = 0 and GTIMER->CCER.CC2P = 0.
- Set the slave mode controller to encoder mode 3 by setting GTIMER->SMCR.SMS[2:0] = 011.

14.4.15 GTIMER Slave Mode

When GTIMER is used as a slave (triggered by an external event), it can be configured to operate in three modes: reset mode, gated mode, and trigger mode.

14.4.15.1 Reset Mode

In this mode, all the preload registers in GTIMER will be reinitialized due to external input events, and the counter will restart counting from 0. The counter counts normally until the rising edge of external TI1 input occurs, at which time the counter is cleared and then restarts counting. The configuration process is as follows:

1. In GPIO module, configure the corresponding pin for GTIMER_CH1 function.
2. Disable the channel by setting GTIMER_CCER.CC1E = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting GTIMER_CCMR1.CC1S = 01.
4. Select the active counting edge by setting GTIMER_CCER.CC1P = 0.
5. Select TI1FP1 as the trigger input signal (TRGI) by setting GTIMER_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to reset mode by setting GTIMER_SMCR.SMS[2:0] = 100.
7. Enable the channel by setting GTIMER_CCER.CC1E = 1.
8. Enable the counter by setting GTIMER_CR1.CEN = 1.

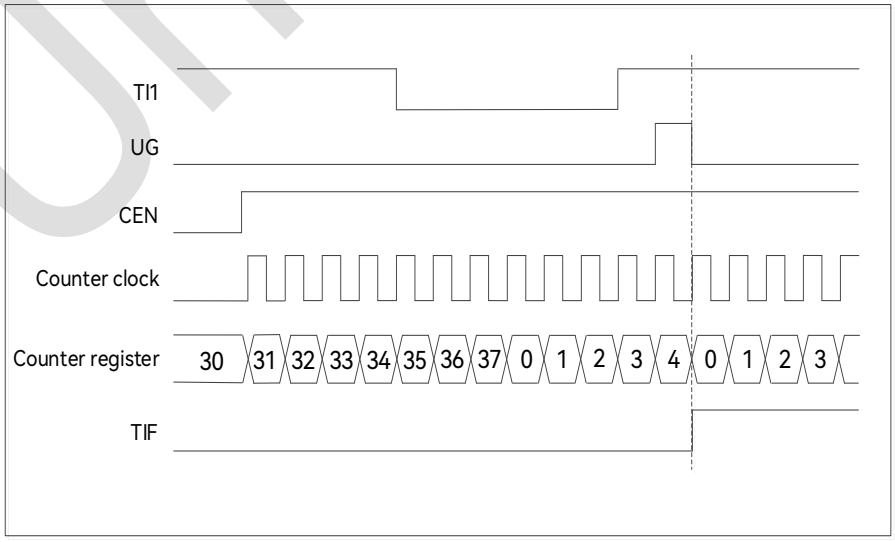


Figure 14-17: Timing Diagram in Reset Mode

14.4.15.2 Gated Mode

In gated mode, the counter operates only when the input signal is at a specific level. Level transitions that cause the counter to start or stop counting will trigger an interrupt flag. The timing configuration process is as follows:

1. In GPIO module, configure the corresponding pin for GTIMER_CH1 function.
2. Disable the channel by setting GTIMER_CCER.CC1E = 0 to ensure the success of subsequent channel configuration.
3. Select the input channel by setting GTIMER_CCMR1.CC1S = 01.
4. Select the active counting edge by setting GTIMER_CCER.CC1P = 1.
5. Select TI1FP1 as the trigger input signal (TRGI) by setting GTIMER_SMCR.TS[2:0] = 101.
6. Configure the slave mode controller to gated mode by setting GTIMER_SMCR.SMS[2:0] = 101.
7. Enable the channel by setting GTIMER_CCER.CC1E = 1.
8. Enable the counter by setting GTIMER_CR1.CEN = 1.

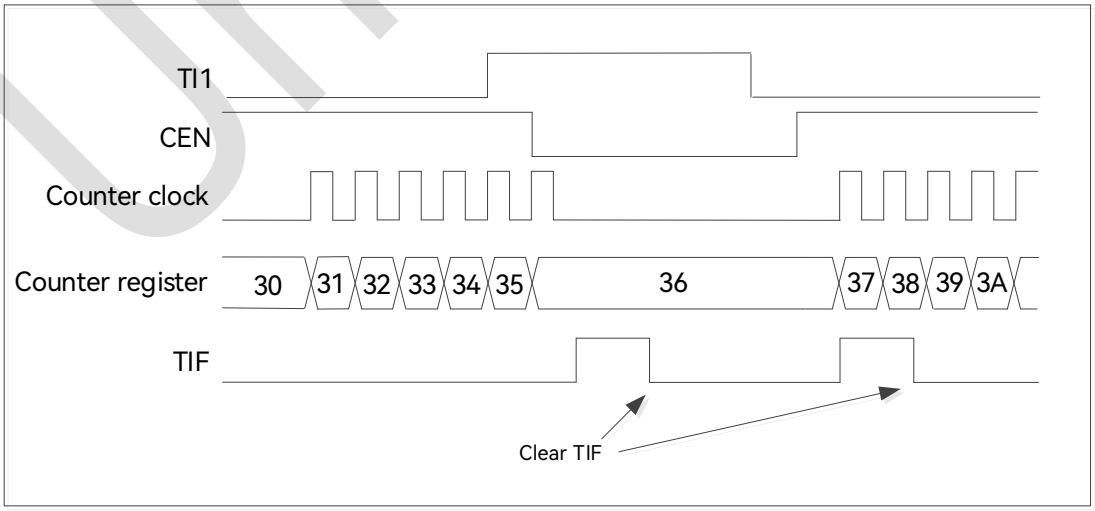


Figure 14-18: Timing Diagram in Gated Mode

14.4.15.3 Trigger Mode

The counter starts counting only after a specific external input event occurs. The timing configuration process is as follows:

1. In GPIO module, configure the corresponding pin for GTIMER_CH1 function.
2. Disable the channel by setting GTIMER->CCER.CC1E = 0 to ensure the success of subsequent channel configurations.
3. Select the input channel by setting GTIMER->CCMR1.CC1S = 01.
4. Select the active counting edge by setting GTIMER->CCER.CC1P = 0.
5. Select TI1FP1 as the trigger input source by setting GTIMER->SMCR.TS[2:0] = 101.
6. Set the slave mode controller to trigger mode by setting GTIMER->SMCR.SMS[2:0] = 110.
7. Enable the channel by setting GTIMER->CCER.CC1E = 1.

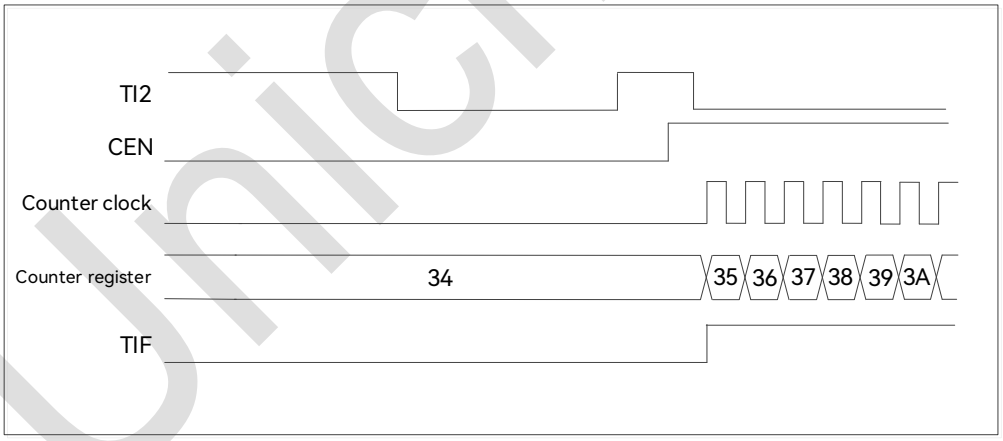


Figure 14-19: Timing Diagram in Trigger Mode

14.4.16 Timer Synchronization

Timers can be cascaded via trigger events to achieve synchronization or cascading. A general-purpose timer can be triggered to count by the outputs of two other general-purpose timers or advanced timers. Meanwhile, the trigger output of a general-purpose timer can be connected

to the internal trigger input of another timer. When a timer is in master mode, it can perform operations such as resetting, starting, stopping, or providing a clock for the counter of another timer in slave mode.

14.4.17 DMA Burst

GTIMER supports DMA and DMA-burst access. It can be configured to trigger a DMA request when a specific event occurs—this allows writing capture results from CCR to RAM, or writing the contents of one or more registers from RAM to the preload register of GTIMER.

DMA-burst allows for multiple successive DMA requests triggered by a single event, primarily for the purpose of continuously updating the contents of multiple registers after an event occurs, thus facilitating dynamic real-time adjustments to output waveforms and other functions.

The DMA controller must point the peripheral target address to a virtual register GTIMER_DMAR. When a specific timer event occurs, GTIMER will continuously issue multiple DMA requests. Each DMA write operation to GTIMER_DMAR will be redirected by GTIMER to the actual functional register.

The DBL register is used to set the DMA burst length, and the DBA register is used to set the base address (relative to the offset of GTIMER_CR) for DMA access to GTIMER.

14.4.18 Input XOR Function

The input signals from channels 1 to 3 can be XORed together and then connected to the input of the filtering and edge detection circuit for input capture or triggering of channel 1. The TI1S bit in the GTIMER_CR2 register is used to select whether the input of channel 1 comes from the XOR of the three channel inputs.

14.4.19 Debug Mode

When the Cortex-M0 enters debug mode, the timer can either stop or continue working, depending on the behavior defined by the STOP_EN register.

14.5 Register Description

14.5.1 GTIMER Registers

GTIMER1 register base address: 0x40003400

GTIMER2 register base address: 0x40003800

Table 14-3: List of GTIMER Registers

Offset Address	Register	Description
0x00	GTIMER_CR1	GTIMER control register 1
0x04	GTIMER_CR2	GTIMER control register 2
0x08	GTIMER_SMCR	GTIMER slave mode control register
0x0C	GTIMER_DIER	GTIMER interrupt and DMA enable register
0x10	GTIMER_SR	GTIMER status register
0x14	GTIMER_EGR	GTIMER event generation register
0x18	GTIMER_CCMR1	GTIMER capture/compare mode register 1
0x1C	GTIMER_CCMR2	GTIMER capture/compare mode register 2
0x20	GTIMER_CCER	GTIMER capture/compare enable register
0x24	GTIMER_CNT	GTIMER counter register
0x28	GTIMER_PSC	GTIMER prescaler register
0x2C	GTIMER_ARR	GTIMER auto-reload register
0x34	GTIMER_CCR1	GTIMER capture/compare register 1
0x38	GTIMER_CCR2	GTIMER capture/compare register 2
0x3C	GTIMER_CCR3	GTIMER capture/compare register 3
0x40	GTIMER_CCR4	GTIMER capture/compare register 4
0x48	GTIMER_DCR	GTIMER DMA control register
0x4C	GTIMER_DMAR	GTIMER DMA access register

14.5.2 Control Register 1 GTIMER_CR1 (Offset: 00H)

Bit	Name	Attribute	Reset Value	Description
31:11	RSV	-	-	Reserved
10	STOP_EN	RW	0x1	Break enable bit: after software sets a breakpoint, the counter is prohibited from counting: 0: Counter operates normally. 1: Counter stops counting.
9:8	CKD	RW	0x0	Digital filter clock frequency division register (division ratio relative to CK_INT): 00: $t_{DTS} = t_{CK\_INT}$ 01: $t_{DTS} = 2 * t_{CK\_INT}$ 10: $t_{DTS} = 4 * t_{CK\_INT}$ 11: Reserved for future use (RFU), disabled
7	ARPE	RW	0x0	Auto-reload preload enable: 0: ARR register does not enable preload. 1: ARR register enables preload.
6:5	CMS	RW	0x0	Counter alignment mode selection: 00: Edge-aligned mode 01: Center-aligned mode 1, where the output compare interrupt flag is set only during down-counting. 10: Center-aligned mode 2, where the output compare interrupt flag is set only during up-counting. 11: Center-aligned mode 3, where the output compare interrupt flag is set during both up-counting and down-counting.
4	DIR	RW	0x0	Counting direction register: 0: Up-counting 1: Down-counting Note: This register is read-only when the timer is configured in center-aligned mode or encoder mode.
3	OPM	RW	0x0	Single-pulse mode output: 0: The counter does not stop at the occurrence of an update event. 1: The counter stops (CEN automatically cleared) at

Bit	Name	Attribute	Reset Value	Description
				the occurrence of an update event.
2	URS	RW	0x0	Update request source: 0: An update interrupt or DMA request can be generated by any of the following events: ● Counter overflow or underflow ● Software setting the UG bit ● Slave controller generating an update 1: An update interrupt or DMA request can be generated only at counter overflow or underflow.
1	UDIS	RW	0x0	Update disable: 0: Enable update event; an update event can be generated by any of the following events: ● Counter overflow or underflow ● Software setting the UG bit ● Slave controller generating an update 1: Disable update event, do not update shadow register. The counter and prescaler will be reinitialized if the UG bit is set or the slave controller receives a hardware reset.
0	CEN	RW	0x0	Counter enable: 0: Counter is disabled. 1: Counter is enabled. Note: External trigger mode can automatically set CEN.

14.5.3 Control Register 2 GTIMER_CR2 (Offset: 04H)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7	TI1S	RW	0x0	GTIMER input TI1 selection: 0: The GTIMER_CH1 pin is connected to TI1 input. 1: The GTIMER_CH1, CH2 and CH3 pins are connected to TI1 input via XOR.
6:4	MMS	RW	0x0	Master mode selection for configuring the source of the

Bit	Name	Attribute	Reset Value	Description
				synchronization trigger signal (TRGO) sent to slave in master mode: 000: The UG bit of GTIMER_EGR is used as TRGO. 001: The counter enable signal CNE_EN is used as TRGO, which can be used to start multiple timers simultaneously. 010: The UE (update event) signal is used as TRGO. 011: Compare pulse, if the CC1IF flag is about to set, TRGO outputs a positive pulse. 100: OC1REF is used as TRGO. 101: OC2REF is used as TRGO. 110: OC3REF is used as TRGO. 111: OC4REF is used as TRGO. Note: The slave timer or ADC must have its working clock enabled in advance to receive the TRGO sent by the master timer.
3	CCDS	RW	0x0	Capture/compare DMA selection: 0: Send DMA request when a capture/compare event occurs. 1: Send DMA request when an update event occurs.
2:0	RSV	-	-	Reserved

14.5.4 Slave Mode Control Register GTIMER_SMCR (Offset: 08H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	ETP	RW	0x0	External trigger signal polarity configuration: 0: Active on high level or rising edge 1: Active on low level or falling edge
14	ECE	RW	0x0	External clock enable: 0: Disable external clock mode 2 1: Enable external clock mode 2, with the counter clocked by any active edge of ETRF
13:12	ETPS	RW	0x0	External trigger signal prescaler register: The frequency of external trigger signal ETRP can be at

Bit	Name	Attribute	Reset Value	Description
				most 1/4 of the GTIMER working clock frequency. When the input signal frequency is high, prescaling can be used. 00: No prescaler 01: Divided by 2 10: Divided by 4 11: Divided by 8
11:8	ETF	RW	0x0	External trigger filter frequency and length selection: 0000: No filter 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS}$ / 2, N = 6 0101: $f_{SAMPLING} = f_{DTS}$ / 2, N = 8 0110: $f_{SAMPLING} = f_{DTS}$ / 4, N = 6 0111: $f_{SAMPLING} = f_{DTS}$ / 4, N = 8 1000: $f_{SAMPLING} = f_{DTS}$ / 8, N = 6 1001: $f_{SAMPLING} = f_{DTS}$ / 8, N = 8 1010: $f_{SAMPLING} = f_{DTS}$ / 16, N = 5 1011: $f_{SAMPLING} = f_{DTS}$ / 16, N = 6 1100: $f_{SAMPLING} = f_{DTS}$ / 16, N = 8 1101: $f_{SAMPLING} = f_{DTS}$ / 32, N = 5 1111: $f_{SAMPLING} = f_{DTS}$ / 32, N = 6 1111: $f_{SAMPLING} = f_{DTS}$ / 32, N = 8
7	MSM	RW	0x0	Master/slave mode selection: 0: No action, to perfectly synchronize the master timer and slave timer via TRGO (when perfectly synchronized, the master register CR2.MMS must be set to 010). 1: Delay the master TRGI trigger signal. Note: This is only applicable to the trigger mode in slave mode (SMS = 110) and is configured by the slave mode.
6:4	TS	RW	0x0	Trigger selection, selecting the trigger source for synchronizing the counter: 000: Internal trigger 0 (ITR0) 001: Internal trigger 0 (ITR1) 010: Internal trigger 0 (ITR2)

Bit	Name	Attribute	Reset Value	Description
				011: Internal trigger 0 (ITR3) 100: TI1 edge detector (TI1F_ED) 101: Filtered timer input 1 (TI1FP1) 110: Filtered timer input 2 (TI2FP2) 111: External trigger input (ETRF) Note: The TS bit can only be rewritten when the slave mode is disabled (i.e. SMS = 000).
3	RSV	-	-	Reserved
2:0	SMS	RW	0x0	Slave mode selection: 000: Slave mode disabled: after CEN is enabled, the prescaler is clocked directly by the internal clock. 001: encoder mode 1—counter counts up/down on TI2FP2 edge depending on TI1FP1 level. 010: Encoder mode 2: the counter counts up/down on the TI1FP1 edge depending on the TI2FP2 level. 011: Encoder mode 3: the counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the levels of other input signals. 100: Reset mode: the rising edge of TRGI initializes the counter and triggers a register update. 101: Gated mode: the counting clock is enabled when TRGI is high, and stopped when TRGI is low. 110: Trigger mode: the counter starts counting at the rising edge of TRGI (the counter does not reset). 111: External clock mode 1: the rising edge of TRGI directly drives the counter.

14.5.5 DMA and Interrupt Enable Register GTIMER_DIER (Offset: 0CH)

Bit	Name	Attribute	Reset Value	Description
31:15	RSV	-	-	Reserved
19	CC4BUEN	RW	0x1	DMA mode configuration for capture/compare

Bit	Name	Attribute	Reset Value	Description
				channel 4: 0: Single transfer, DMA reads the value of the CCR4 register and transfers it to RAM. 1: Burst mode from RAM to GTIMER, with access address and length configured via DCR.
18	CC3BUEN	RW	0x1	DMA mode configuration for capture/compare channel 3: 0: CCR3 to RAM, DMA reads the value of the CCR3 register and transfers it to RAM. 1: Burst mode from RAM to GTIMER, with access address and length configured via DCR.
17	CC2BUEN	RW	0x1	DMA mode configuration for capture/compare channel 2: 0: CCR2 to RAM, DMA reads the value of the CCR2 register and transfers it to RAM. 1: Burst mode from RAM to GTIMER, with access address and length configured via DCR.
16	CC1BUEN	RW	0x1	DMA mode configuration for capture/compare channel 1: 0: CCR1 to RAM, DMA reads the value of the CCR1 register and transfers it to RAM. 1: Burst mode from RAM to GTIMER, with access address and length configured via DCR.
15	RSV	-	-	Reserved
14	TDE	RW	0x0	External trigger DMA request enable: 0: In slave mode, disable DMA requests generated by external trigger events. 1: In slave mode, enable DMA requests generated by external trigger events (can be used to automatically update the preload register).
13	RSV	-	-	Reserved
12	CC4DE	RW	0x0	Capture/compare channel 4 DMA request enable: 0: Disable CC4 DMA request. 1: Enable CC4 DMA request.
11	CC3DE	RW	0x0	Capture/compare channel 3 DMA request enable: 0: Disable CC3 DMA request.

Bit	Name	Attribute	Reset Value	Description
				1: Enable CC3 DMA request.
10	CC2DE	RW	0x0	Capture/compare channel 2 DMA request enable: 0: Disable CC2 DMA request. 1: Enable CC2 DMA request.
9	CC1DE	RW	0x0	Capture/compare channel 1 DMA request enable: 0: Disable CC1 DMA request. 1: Enable CC1 DMA request.
8	UDE	RW	0x0	Update event DMA request enable: 0: Disable DMA request generation on update event occurrence. 1: Enable DMA request generation on update event occurrence.
7	RSV	-	-	Reserved
6	TIE	RW	0x0	Trigger event interrupt enable: 0: Disable trigger event interrupt. 1: Enable trigger event interrupt.
5	RSV	-	-	Reserved
4	CC4IE	RW	0x0	Capture/compare channel 4 interrupt enable: 0: Disable capture/compare channel 4 interrupt. 1: Enable capture/compare channel 4 interrupt.
3	CC3IE	RW	0x0	Capture/compare channel 3 interrupt enable: 0: Disable capture/compare channel 3 interrupt. 1: Enable capture/compare channel 3 interrupt.
2	CC2IE	RW	0x0	Capture/compare channel 2 interrupt enable: 0: Disable capture/compare channel 2 interrupt. 1: Enable capture/compare channel 2 interrupt.
1	CC1IE	RW	0x0	Capture/compare channel 1 interrupt enable: 0: Disable capture/compare channel 1 interrupt. 1: Enable capture/compare channel 1 interrupt.
0	UIE	RW	0x0	Update event interrupt enable: 0: Disable update event interrupt. 1: Enable update event interrupt.

14.5.6 Status Register GTIMER_SR (Offset: 10H)

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12	CC4OF	W1C	0x0	Capture/compare channel 4 overcapture status: Refer to CC1OF.
11	CC3OF	W1C	0x0	Capture/compare channel 3 overcapture status: Refer to CC1OF.
10	CC2OF	W1C	0x0	Capture/compare channel 2 overcapture status: Refer to CC1OF.
9	CC1OF	W1C	0x0	Capture/compare channel 1 overcapture status: This register is only valid when the corresponding channel is configured in input capture mode. This bit is set by hardware and cleared by software writing 1. 0: No overcapture event 1: A new capture occurs while the CC1IF flag is 1.
8:7	RSV	-	-	Reserved
6	TIF	W1C	0x0	Trigger interrupt flag, set by hardware and cleared by software writing 1.
5	RSV	-	-	Reserved
4	CC4IF	W1C	0x0	Capture/compare channel 4 interrupt flag: Refer to CC1IF.
3	CC3IF	W1C	0x0	Capture/compare channel 3 interrupt flag: Refer to CC3IF.
2	CC2IF	W1C	0x0	Capture/compare channel 2 interrupt flag: Refer to CC2IF.
1	CC1IF	W1C	0x0	Capture/compare channel 1 interrupt flag: If CC1 channel is configured for output: CC1IF is set when the counter value equals the compare value, and cleared by software writing 1. If CC1 channel is configured for input: CC1IF is set when a capture event occurs, and cleared by software writing 1, or automatically cleared by software reading CCR1.
0	UIF	W1C	0x0	Update event interrupt flag, set by hardware and cleared by software writing 1. UIF is set and the shadow register is updated at the

Bit	Name	Attribute	Reset Value	Description
				following events: Counter overflow occurs if repetition counter = 0 and UDIS = 0. The counter is reinitialized by software setting the UG bit if URS = 0 and UDIS = 0. The counter is reinitialized by a trigger event if URS = 0 and UDIS = 0.

14.5.7 Event Generation Register GTIMER_EGR (Offset: 14H)

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	TG	W	0x0	Software trigger: This bit is set by software to generate a trigger event, and automatically cleared by hardware.
5	RSV	-	-	Reserved
4	CC4G	W	0x0	Capture/compare channel 4 software trigger, refer to CC1G
3	CC3G	W	0x0	Capture/compare channel 3 software trigger, refer to CC1G
2	CC2G	W	0x0	Capture/compare channel 2 software trigger, refer to CC1G
1	CC1G	W	0x0	Capture/compare channel 1 software trigger, automatically cleared by hardware: If CC1 channel is configured for output: CC1IF is set, and if enabled, it can generate the corresponding interrupt and DMA request. If CC1 channel is configured for input: the current counter value is captured into the CCR1 register, CC1IF is set, and if enabled, it can generate the corresponding interrupt and DMA request.
0	UG	W	0x0	Software update event: This bit can be set by software to generate an update event, and is automatically cleared by hardware. When the software sets UG, the counter will be

Bit	Name	Attribute	Reset Value	Description
				reinitialized, the shadow register will be updated, and the prescaler counter will be cleared.

14.5.8 Capture/Compare Mode Register 1 GTIMER_CCMR1 (Offset: 18H)

This register is multiplexed for two different functions under output compare and input capture modes:

1. Output compare mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	OC2CE	RW	0x0	Output compare 2 clear enable, refer to OC1CE
14:12	OC2M	RW	0x0	Output compare 2 mode configuration, refer to OC1M
11	OC2PE	RW	0x0	Output compare 2 preload enable, refer to OC1PE
10	OC2FE	RW	0x0	Output compare 2 fast enable, refer to OC1FE
9:8	CC2S	RW	0x0	Capture/compare channel 2 selection: 00: CC2 channel is configured as output. 01: CC2 channel is configured as input, IC2 is mapped on TI2. 10: CC2 channel is configured as input, IC2 is mapped on TI1. 11: CC2 channel is configured as input, IC2 is mapped on TRC (mapped on TI2 edge detection if BPS2 is set to 1). Note: CC2S can only be written when the channel is disabled (CC2E = 0).
7	OC1CE	RW	0x0	Output compare 1 clear enable: 0: OC1REF is not affected by ETRF. 1: OC1REF is automatically cleared when ETRF is high.
6:4	OC1M	RW	0x0	Output compare 1 mode configuration: these bits define the behavior of the OC1REF signal.

Bit	Name	Attribute	Reset Value	Description
				000: The comparison result between the output compare register CCR1 and the counter CNT does not affect the output. 001: OC1REF is set high when CCR1 = CNT. 010: OC1REF is set low when CCR1 = CNT. 011: OC1REF is toggled when CCR1 = CNT. 100: OC1REF is fixed low (inactive). 101: OC1REF is fixed high (active). 110: PWM mode 1—In up-counting, OC1REF is set high when CNT < CCR1, otherwise set low; in down-counting, OC1REF is set low when CNT > CCR1, otherwise set high. 111: PWM mode 2—In up-counting, OC1REF is set low when CNT < CCR1, otherwise set high; in down-counting, OC1REF is set high when CNT > CCR1, otherwise set low.
3	OC1PE	RW	0x0	Output compare 1 preload enable: 0: The CCR1 preload register is disabled, and CCR1 can be written directly. 1: The CCR1 preload register is enabled, and read/write operations on CCR1 access the preload register, with content shifted to the shadow register only when an update event occurs.
2	OC1FE	RW	0x0	Output compare 1 fast enable: 0: Disable fast enable, the trigger input will not affect the comparison output. 1: Enable fast enable, the trigger input will immediately change OC1REF to the output corresponding to the compare match, regardless of the actual current comparison status. Note: This function acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC1S	RW	0x0	Capture/compare channel 1 selection: 00: CC1 channel is configured as output. 01: CC1 channel is configured as input, IC1 is mapped on TI1.

Bit	Name	Attribute	Reset Value	Description
				10: CC1 channel is configured as input, IC1 is mapped on TI2. 11: CC1 channel is configured as input, IC1 is mapped on TRC (mapped on TI1 edge detection if BPS1 is set to 1). Note: CC1S can only be written when the channel is disabled (CC1E = 0).

2. Input capture mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	IC2F	RW	0x0	Input capture 2 filter
11:10	IC2PSC	RW	0x0	Input capture 2 prescaler
9:8	CC2S	RW	0x0	Capture/compare channel 2 selection: 00: CC2 channel is configured as output. 01: CC2 channel is configured as input, IC2 is mapped on TI2. 10: CC2 channel is configured as input, IC2 is mapped on TI1. 11: CC2 channel is configured as input, IC2 is mapped on TRC (if BPS2 is set to 1, then mapped on TI2 edge detection). Note: CC2S can only be written when the channel is disabled (CC2E = 0).
7:4	IC1F	RW	0x0	Input capture 1 filter: This bit-field defines the sampling frequency and filter length for TI1. 0000: No filter, sampling at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6

Bit	Name	Attribute	Reset Value	Description
				0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
3:2	IC1PSC	RW	0x0	Input capture 1 prescaler: 00: No prescaler 01: Capture occurs once every 2 event inputs 10: Capture occurs once every 4 event inputs 11: Capture occurs once every 8 event inputs Note: The IC1PSC register is reset when CC1E = 0.
1:0	CC1S	RW	0x0	Capture/compare channel 1 selection: 00: CC1 channel is configured as output. 01: CC1 channel is configured as input, IC1 is mapped on TI1. 10: CC1 channel is configured as input, IC1 is mapped on TI2. 11: CC1 channel is configured as input, IC1 is mapped on TRC (mapped on TI1 edge detection if BPS1 is set to 1). Note: CC1S can only be written when the channel is disabled (CC1E = 0).

14.5.9 Capture/Compare Mode Register 2 GTIMER_CCMR2 (Offset: 1CH)

This register is multiplexed for two different functions under output compare and input capture modes.

1. Output compare mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15	OC4CE	RW	0x0	Output compare 4 clear enable, refer to OC1CE
14:12	OC4M	RW	0x0	Output compare 4 mode configuration, refer to OC1M
11	OC4PE	RW	0x0	Output compare 4 preload enable, refer to OC1PE
10	OC4FE	RW	0x0	Output compare 4 fast enable, refer to OC1FE
9:8	CC4S	RW	0x0	Capture/compare channel 4 selection: 00: CC4 channel is configured as output. 01: CC4 channel is configured as input, IC4 is mapped on TI4. 10: CC4 channel is configured as input, IC4 is mapped on TI3. 11: CC4 channel is configured as input, IC4 is mapped on TRC (mapped on TI4 edge detection if BPS4 is set to 1). Note: CC4S can only be written when the channel is disabled (CC4E = 0).
7	OC3CE	RW	0x0	Output compare 4 clear enable: 0: OC4REF is not affected by ETRF. 1: OC4REF is automatically cleared when ETRF is high.
6:4	OC3M	RW	0x0	Output compare 3 mode: these bits define the behavior of the OC3REF signal. 000: The comparison result between the output compare register CCR3 and the counter CNT does not affect the output. 001: OC3REF is set high when CCR3 = CNT. 010: OC3REF is set low when CCR3 = CNT. 011: OC3REF is toggled when CCR3 = CNT. 100: OC3REF is fixed low (inactive). 101: OC3REF is fixed high (active). 110: PWM mode 1—In up-counting, OC3REF is set high when CNT < CCR3, otherwise set low; in down-counting, OC3REF is set low when CNT > CCR3, otherwise set high. 111: PWM mode 2—In up-counting, OC3REF is set low

Bit	Name	Attribute	Reset Value	Description
				when CNT < CCR3, otherwise set high; in down-counting, OC3REF is set high when CNT > CCR3, otherwise set low.
3	OC3PE	RW	0x0	Output compare 3 preload enable: 0: The CCR3 preload register is disabled; and CCR3 can be written directly. 1: The CCR3 preload register is enabled, and read/write operations on CCR3 access the preload register, with content shifted to the shadow register only when an update event occurs.
2	OC3FE	RW	0x0	Output compare 3 fast enable: 0: Disable fast enable, the trigger input will not affect the comparison output. 1: Enable fast enable, the trigger input will immediately change OC3REF to the output corresponding to the compare match, regardless of the actual current comparison status. Note: This function acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC3S	RW	0x0	Capture/compare channel 3 selection: 00: CC3 channel is configured as output. 01: CC3 channel is configured as input, IC3 is mapped on TI3. 10: CC3 channel is configured as input, IC3 is mapped on TI4. 11: CC3 channel is configured as input, IC3 is mapped on TRC (mapped on TI3 edge detection if BPS3 is set to 1). Note: CC3S can only be written when the channel is disabled (CC3E = 0).

2. Input capture mode

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	IC4F	RW	0x0	Input capture 4 filter

Bit	Name	Attribute	Reset Value	Description
11:10	IC4PSC	RW	0x0	Input capture 4 prescaler
9:8	CC4S	RW	0x0	Capture/compare channel 4 selection: 00: CC4 channel is configured as output. 01: CC4 channel is configured as input, IC4 is mapped on TI4. 10: CC4 channel is configured as input, IC4 is mapped on TI3. 11: CC4 channel is configured as input, IC4 is mapped on TRC (mapped on TI4 edge detection if BPS4 is set to 1). Note: CC4S can only be written when the channel is disabled (CC4E = 0).
7:4	IC3F	RW	0x0	Input capture 3 filter: This bit-field defines the sampling frequency and filter length for TI3. 0000: No filter, sampling at f_{DTS} 0001: $f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010: $f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011: $f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8
3:2	IC3PSC	RW	0x0	Input capture 3 prescaler: 00: No prescaler 01: Capture occurs once every 2 event inputs 10: Capture occurs once every 4 event inputs. 11: Capture occurs once every 8 event inputs. Note: The IC3PSC register is reset when CC3E = 0.

Bit	Name	Attribute	Reset Value	Description
1:0	CC3S	RW	0x0	Capture/compare channel 3 selection: 00: CC3 channel is configured as output. 01: CC3 channel is configured as input, IC3 is mapped on TI3. 10: CC3 channel is configured as input, IC3 is mapped on TI4. 11: CC3 channel is configured as input, IC3 is mapped on TRC (mapped on TI3 edge detection if BPS3 is set to 1). Note: CC3S can only be written when the channel is disabled (CC1E = 0).

14.5.10 Capture/Compare Enable Register GTIMER_CCER (Offset: 20H)

Bit	Name	Attribute	Reset Value	Description
31:14	RSV	-	-	Reserved
13	CC4P	RW	0x0	Capture/compare 4 output polarity: refer to CC1P
12	CC4E	RW	0x0	Capture/compare 4 output enable: refer to CC1E
11:10	RSV	-	-	Reserved
9	CC3P	RW	0x0	Capture/compare 3 output polarity: refer to CC1P
8	CC3E	RW	0x0	Capture/compare 3 output enable: refer to CC1E
7:6	RSV	-	-	Reserved
5	CC2P	RW	0x0	Capture/compare 2 output polarity: refer to CC1P
4	CC2E	RW	0x0	Capture/compare 2 output enable: refer to CC1E
3:2	RSV	-	-	Reserved
1	CC1P	RW	0x0	Capture/compare 1 output polarity: When CC1 channel is configured as output: 0: OC1 is active high. 1: OC1 is active low. When CC1 channel is configured as input: 0: Non-inverted mode: capture occurs on the rising edge of IC1. 1: Inverted mode: capture occurs on the falling edge of IC1.

Bit	Name	Attribute	Reset Value	Description
0	CC1E	RW	0x0	Capture/compare 1 output enable: When CC1 channel is configured as output: 0: OC1 is not active. 1: OC1 is active. When CC1 channel is configured as input: 0: Capture function is disabled. 1: Capture function is enabled.

14.5.11 Counter Register GTIMER_CNT (Offset: 24H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CNT	RW	0x0	Counter value

14.5.12 Prescaler Register GTIMER_PSC (Offset: 28H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	PSC	RW	0x0	The counter clock (CK_CNT) frequency is: $f_{CK\_CNT} = f_{CK\_PSC} / (PSC[15:0] + 1)$ This is a preload register whose contents are transferred into the shadow register at each update event. Note: The maximum supported PWM output frequency is 30 MHz, which shall be taken into account when configuring PSC and ARR.

14.5.13 Auto-reload Register GTIMER_ARR (Offset: 2CH)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	ARR	RW	0x0	Auto-reload value at counter overflow This is a preload register whose contents are transferred into the shadow register at each update event.

14.5.14 Capture/Compare Register 1 GTIMER_CCR1 (Offset: 34H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR1	RW	0x0	Capture/compare channel 1 register **If channel CC1 is configured as output:** This is a preload register whose contents are loaded into the shadow register and then compare with the counter to generate the OC1 output. **If channel CC1 is configured as input:** CCR1 stores the counter value at the time of the most recent input capture event, at which point CCR1 is read-only.

14.5.15 Capture/Compare Register 2 GTIMER_CCR2 (Offset: 38H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR2	RW	0x0	Capture/compare channel 2 register **If channel CC2 is configured as output:** This is a preload register whose contents are loaded into the shadow register and then compared with the counter to generate the OC2 output. **If channel CC2 is configured as input:** CCR2 stores the counter value at the time of the most recent input capture event, at which point CCR2 is read-only.

14.5.16 Capture/Compare Register 3 GTIMER_CCR3 (Offset: 3CH)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR3	RW	0x0	Capture/compare channel 3 register **If channel CC3 is configured as output:** This is a preload register whose contents are

Bit	Name	Attribute	Reset Value	Description
				loaded into the shadow register and then compared with the counter to generate the OC3 output. **If channel CC3 is configured as input:** CCR3 stores the counter value at the time of the most recent input capture event, at which point CCR3 is read-only.

14.5.17 Capture/Compare Register 4 GTIMER_CCR4 (Offset: 40H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	CCR4	RW	0x0	Capture/compare channel 4 register **If channel CC4 is configured as output:** This is a preload register whose contents are loaded into the shadow register and then compared with the counter to generate the OC4 output. **If channel CC4 is configured as input:** CCR4 stores the counter value at the time of the most recent input capture event, at which point CCR4 is read-only.

14.5.18 DMA Control Register GTIMER_DCR (Offset: 48H)

Bit	Name	Attribute	Reset Value	Description
31:13	RSV	-	-	Reserved
12:8	DBL	RW	0x0	DMA burst length: Reading from or writing to the MAR register will trigger a burst DMA operation, with the burst length ranging from 1 to 18. 00000: Length = 1 00001: Length = 2 …… 10001: Length = 18 Others: Invalid values, writing is prohibited.

Bit	Name	Attribute	Reset Value	Description
7:5	RSV	-	-	Reserved
4:0	DBA	RW	0x0	DMA base address, defining the offset address directed to the register: 00000: CR1 00001: CR2 00010: SMCR …… 01011: ARR 01100: RSV 01101: CCR1 …… 10000: CCR4 10001: RSV 11000: DCR Note: When DBA + DBL exceeds the GTIMER register address range, the actual burst transfer stops automatically when it reaches the highest GTIMER register address, i.e., the burst length is shortened.

14.5.19 DMA Access Register GTIMER_DMAR (Offset: 4CH)

Bit	Name	Attribute	Reset Value	Description
31:0	DMAR	RW	0x400	DMA burst access register: When using DMA burst transfer, set the DMA channel peripheral address to DMAR, and GTIMER will generate multiple DMA requests based on the value of DBL.

15 LPTIMER

15.1 Overview

The LPTIMER is a 32-bit low-power timer/counter module. Due to the variety of its clock sources, it can remain operational with extremely low power consumption. The LPTIMER can operate without an internal clock, enabling external pulse counting in sleep mode. It can also work in conjunction with external trigger signals to achieve low-power timeout wake-up.

15.2 Main Features

- Independent 32-bit up counter
- 3-bit asynchronous clock prescaler with 8 division factors (1, 2, 4, 8, 16, 32, 64, 128)
- Selectable operating clocks:
  - Internal clock sources: LSCLK (CLK32K), RCLP (LVD), PCLK
  - External clock sources: LPTIMER_IN (with analog filtering), COMP_IN (COMP0–COMP1)
- 32-bit compare/capture register
- 32-bit target value register
- Continuous/single-trigger mode
- Configurable input polarity
- External pulse counting without clock
- Sleep timeout wake-up triggered by external signal
- PWM output

15.3 Block Diagram

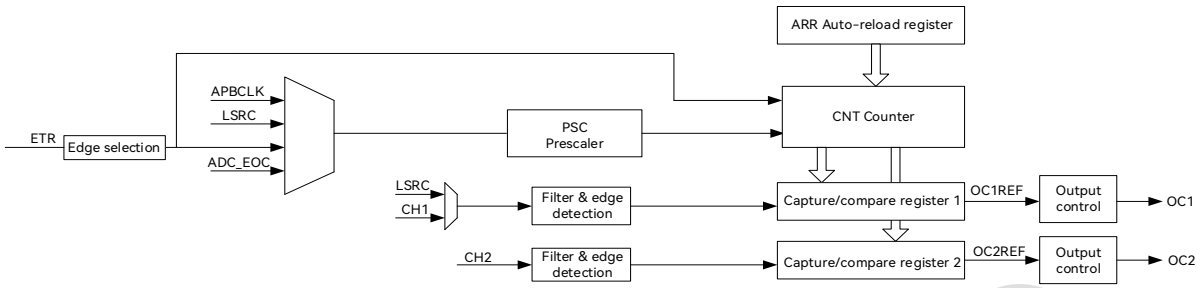


Figure 15-1: LPTIMER Block Diagram

15.4 Timer Function

LPTIMER operating modes: counting, ETR trigger counting, ETR pulse counting, and timeout mode. The software flow for LPTIMER in counting mode is as follows:

- Select the operating clock.
- Set the ARR register.
- Configure the EN enable bit in CR1.
- Wait for the counter value to equal ARR, generating UIF interrupt and resetting the counter value to 0.

15.4.1 Single-shot Counting and Continuous Counting

LPTIMER has two counting modes: single-shot counting and continuous counting.

Single-shot counting mode: After being triggered, the counter counts up to ARR, returns to 0, and stops automatically, generating an overflow interrupt, while the EN bit is automatically cleared by hardware.

Continuous counting mode: The counter remains running once started until it is disabled. When

the counter reaches ARR, it returns to 0 and restarts counting, generating an overflow interrupt.

15.4.2 ETR Trigger Counting

In the external ETR trigger counting mode, LPTIMER uses the signal input from the LPTIMER_ETR pin as the trigger signal. The LPTIMER_ETR signal is first sampled and synchronized by the LPTIMER operating clock, and the timer can be incremented on its rising edge, falling edge, or both edges.

15.4.3 ETR Pulse Counting

In the external ETR pulse counting mode, LPTIMER directly uses the signal input from the ETR pin as the counting clock. The software can select the counting edge and prescaler value.

Note: Unlike the ETR trigger counting mode, in ETR trigger counting operation, if the trigger clock is extremely fast and not sampled by the LPTIMER operating clock, the ETR trigger counting mode will not count. In contrast, the ETR pulse counting mode does not require sampling of the trigger signal and can count using the signal as the operating clock.

15.4.4 Timeout Mode

In timeout mode, the LPTIMER uses the signal input from the ETR pin as the trigger signal, and the timer operates with the internal clock CLK_LPT. After the timer is enabled in timeout mode, it waits for the first valid edge of the ETR signal to start counting. Subsequently, each new valid edge of the ETR signal will reset the counter and restart counting. If the timer overflows, it generates an overflow interrupt, resets the count value to 0, and automatically clears the EN bit to end the counting process.

15.5 Capture/Compare Function

LPTIMER features two channels for capture and compare functions. When used in capture mode, it can capture the period length of external or internal signals. When used in compare function, it can output corresponding PWM waveforms based on different comparison values.

15.5.1 PWM Output

Both capture/compare channels of the LPTIMER can output 32-bit PWM waveforms. The PWM function requires configuring the capture/compare channel for compare output. After enabling the PWM function, the LPTIMER starts counting from 0. The output goes high when the count value equals CCR, and goes low when the count value equals ARR. The PWM period is determined by the ARR register, while the duty cycle is determined by the CCRx register. Software can configure the polarity of the output waveform.

15.5.2 Input Capture

The input capture of the LPTIMER can be configured to capture on the rising edge, falling edge, or both edges of the input signal. The captured CNT value is stored in the CCRx register and a capture interrupt is generated.

15.5.3 Debug Mode

When the Cortex-M0 enters debug mode, the timer can either stop or continue working, with its behavior defined by the STOP_EN register.

15.6 Register Description

LPTIMER register base address: 0x40001000

Table 15-1: List of LPTIMER Registers

Offset	Name	Description
0x00	LPTIMER_CR1	LPTIMER control register 1
0x04	LPTIMER_CR2	LPTIMER control register 2
0x08	LPTIMER_IER	LPTIMER interrupt enable register
0x0C	LPTIMER_SR	LPTIMER interrupt flag register
0x10	LPTIMER_CNT	LPTIMER counter value register
0x14	LPTIMER_CCMCFG1	LPTIMER capture/compare configuration register 1
0x18	LPTIMER_CCMCFG2	LPTIMER capture/compare configuration register 2
0x1C	LPTIMER_ARR	LPTIMER auto-reload register 1
0x20	LPTIMER_CCR1	LPTIMER capture/compare register 1
0x24	LPTIMER_CCR2	LPTIMER capture/compare register 2
0x28	LPTIMER_LOAD	LPTIMER counter value load register 1
0x2C	LPTIMER_BUFFER	LPTIMER counter buffer register 1

15.6.1 LPTIMER Control Register 1 LPTIMER_CR1 (Offset: 00H)

Bit	Name	Attribute	Default Value	Functional Description
31:5	RSV	-	-	Reserved
4	STOP_EN	RW	0x1	LPTIMER break enable bit: after software sets a breakpoint, the counter is prohibited from counting: 0: Counter operates normally. 1: Counter stops counting.
3:1	RSV	-	-	Reserved
0	EN	RW	0x0	LPTIMER enable: 1: Enable counter counting. 0: Disable counter counting.

15.6.2 LPTIMER Control Register 2 LPTIMER_CR2 (Offset: 04H)

Bit	Name	Attribute	Default Value	Functional Description
31:15	RSV	-	-	Reserved

Bit	Name	Attribute	Default Value	Functional Description
14:12	CAP1SRSEL	RW	0x0	Channel 1 capture signal source selection: 000: LPTIMER _CH1 input 001: RCLP 010: UART0_RX 011: UART1_RX 100: I2C_SCL 101: SPI_SCK 110: COMP0_IN1 111: COMP1_IN1
11:9	DIVCFG	RW	0x0	Counter clock division factor selection: 000: Divided by 1 001: Divided by 2 010: Divided by 4 011: Divided by 8 100: Divided by 16 101: Divided by 32 110: Divided by 64 111: Divided by 128
8:6	CLKSEL	RW	0x0	Clock source selection: 000: LSCLK (system low-speed clock CLK32K selected by SYSREG->SYSCTRL0[12], i.e., RCL or XTL) as the counting clock 001: RCLP as the counting clock (LVD) 010: Gated PCLK as the counting clock 011: ETR input as the counting clock 100: COMP0_IN 101: COMP1_IN Other: COMP0_IN
5	EDGESEL	RW	0x0	ETR counting clock input edge selection: 0: Count on the rising edge of ETR. 1: Count on the falling edge of ETR.

Bit	Name	Attribute	Default Value	Functional Description
4	SINGLE	RW	0x0	Single-shot counting mode enable: 0: Continuous counting mode: The counter keeps running after being triggered until it is stopped. Upon reaching the target value, the counter restart counting from 0, generating an overflow interrupt. 1: Single-shot counting mode: After being triggered, the counter counts to the target value, then returns to 0 and automatically stops, generating an overflow interrupt.
3:2	TRIGEDGE	RW	0x0	External trigger edge selection: 00: Trigger on the rising edge of the ETR input signal 01: Trigger on the falling edge of the ETR input signal 10/11: Trigger on both rising and falling edges of the ETR input signal
1:0	LPMOD	RW	0x0	Operating mode selection: 00: Counting mode 01: ETR pulse-trigger counting mode 10: ETR pulse counting mode 11: Timeout mode

15.6.3 LPTIMER Interrupt Enable Register LPTIMER_IER (Offset: 08H)

Bit	Name	Attribute	Default Value	Functional Description
31:4	RSV	-	-	Reserved
3	TIE	RW	0x0	LPTIMER trigger event interrupt enable: 0: Disable trigger event interrupt. 1: Enable trigger event interrupt.
2	CC2IE	RW	0x0	LPTIMER capture/compare channel 2 interrupt enable: 0: Disable capture/compare channel 2 interrupt. 1: Enable capture/compare channel 2 interrupt.

Bit	Name	Attribute	Default Value	Functional Description
1	CC1IE	RW	0x0	LPTIMER capture/compare channel 1 interrupt enable: 0: Disable capture/compare channel 1 interrupt. 1: Enable capture/compare channel 1 interrupt.
0	UIE	RW	0x0	LPTIMER update event interrupt enable: 0: Disable update event interrupt. 1: Enable update event interrupt.

15.6.4 LPTIMER Interrupt Flag Register LPTIMER_SR (Offset: 0CH)

Bit	Name	Attribute	Default Value	Functional Description
31:10	RSV	-	-	Reserved
9	LPTIM_CH2	R	0x0	LPTIMER capture channel 2 level status
8	LPTIM_CH1	R	0x0	LPTIMER capture channel 1 level status
7:4	RSV	-	-	Reserved
3	TIF	W1C	0x0	LPTIMER trigger interrupt flag, set by hardware and cleared by software writing 1: 0: No trigger event 1: Trigger interrupt pending
2	CC2IF	W1C	0x0	LPTIMER capture/compare channel 2 interrupt flag, set by hardware and cleared by software writing 1: 0: No compare or capture interrupt generated 1: Interrupt generated when the CNT value equals CCR2, or when an input capture event occurs.
1	CC1IF	W1C	0x0	LPTIMER capture/compare channel 1 interrupt flag, set by hardware and cleared by software writing 1: 0: No compare or capture interrupt generated 1: Interrupt generated when the CNT value equals CCR1, or when an input capture event occurs.
0	UIF	W1C	0x0	LPTIMER update event interrupt flag, set by hardware and cleared by software writing 1:

Bit	Name	Attribute	Default Value	Functional Description
				0: No update event interrupt generated 1: Interrupt generated when the CNT value equals ARR

15.6.5 LPTIMER Counter Value Register LPTIMER_CNT (Offset: 10H)

Bit	Name	Attribute	Default Value	Functional Description
31:0	CNT	R	0x0	Counter value

15.6.6 LPTIMER Capture/Compare Configuration Register 1 LPTIMER_CCMCFG1 (Offset: 14H)

Bit	Name	Attribute	Default Value	Functional Description
31:6	RSV	-	-	Reserved
5:4	CAP1EDGE	RW	0x0	Channel 1 capture edge selection: 00: Capture on rising edge 01: Capture on falling edge 10: Capture on both rising and falling edges 11: Undefined
3	RSV	-	-	Reserved
2	CC1P	RW	0x0	Channel 1 compare output waveform polarity selection: 0: Output is low when CNT ≤ CCR1 and high when CNT > CCR1. 1: Output is high when CNT ≤ CCR1 and low when CNT > CCR1.
1	CC1S	RW	0x0	Channel 1 capture/compare output selection: 0: Channel 1 is configured for output. 1: Channel 1 is configured for input.
0	CC1E	RW	0x0	Channel 1 capture enable: 0: Channel 1 capture function is disabled. 1: Channel 1 capture function is enabled.

15.6.7 LPTIMER Capture/Compare Configuration Register 2 LPTIMER_CCMCFG2 (Offset: 18H)

Bit	Name	Attribute	Default Value	Functional Description
31:6	RSV	-	-	Reserved
5:4	CAP2EDGE	RW	0x0	Channel 2 capture edge selection: 00: Capture on rising edge 01: Capture on falling edge 10: Capture on both rising and falling edges 11: Undefined
3	RSV	-	-	Reserved
2	CC2P	RW	0x0	Channel 2 output polarity selection: 0: Output is low when CNT ≤ CCR2 and high when CNT > CCR2. 1: Output is high when CNT ≤ CCR2 and low when CNT > CCR2.
1	CC2S	RW	0x0	Channel 2 capture/compare selection: 0: Channel 2 is configured for output. 1: Channel 2 is configured for input.
0	CC2E	RW	0x0	Channel 2 capture enable: 0: Channel 2 capture function is disabled. 1: Channel 2 capture function is enabled.

15.6.8 LPTIMER Auto-reload Register LPTIMER_ARR (Offset: 1CH)

Bit	Name	Attribute	Default Value	Functional Description
31:0	ARR	RW	0x0	Auto-reload target register: When the counter value is equal to the ARR value, the counter returns to 0. Note: The maximum supported PWM output frequency is 12 MHz, which shall be taken into account when configuring DIVCFG and ARR.

15.6.9 LPTIMER Capture/Compare Register 1 LPTIMER_CCR1 (Offset: 20H)

Bit	Name	Attribute	Default Value	Functional Description
31:0	CCR1	RW	0x0	Capture/compare value register 1: If ARR = CCR1, CCR1 shall prevail.

15.6.10 LPTIMER Capture/Compare Register 2 LPTIMER_CCR2 (Offset: 24H)

Bit	Name	Attribute	Default Value	Functional Description
31:0	CCR2	RW	0x0	Capture/compare value register 2: If ARR = CCR2, CCR2 shall prevail.

15.6.11 LPTIMER Counter Value Load Register LPTIMER_LOAD (Offset: 28H)

Bit	Name	Attribute	Default Value	Functional Description
31:1	RSV	-	-	Reserved
0	LOAD	RW	0x0	1: The counter loading is in progress. 0: The counter loading has been completed.

15.6.12 LPTIMER Counter Buffer Register LPTIMER_BUFFER (Offset: 2CH)

Bit	Name	Attribute	Default Value	Functional Description
31:0	BUFFER	R	0x0	Counter buffer register: This register is used to store the current value of the counter after the software issues a load command.

15.7 Operation Procedure

The following details the software usage flow.

15.7.1 General-purpose Timer

1. Initialize the LPTIMER clock module.
2. Configure LPTIMER_CR2.LPMOD to select the operating mode.
3. Configure LPTIMER_CR2.SINGLE to set the counting mode.
4. Configure LPTIMER_CR2.DIVCFG to set the division factor.
5. Configure LPTIMER_CR2.CLKSEL to set the clock source.
6. Set the target value in the LPTIMER_ARR1 register.
7. Enable the LPTIMER_IER interrupt register to select the overflow interrupt.
8. Enable LPTIMER_CR1.EN to start the counter.

15.7.2 Input Capture Function with DMA

1. Initialize the LPTIMER clock module.
2. Configure LPTIMER_CR2.SINGLE to set the counting mode.
3. Configure LPTIMER_CR2.DIVCFG to set the division factor.
4. Configure LPTIMER_CR2.CLKSEL to set the clock source.
5. Configure LPTIMER_CR2.TRIGEDGE to set the capture edge for external input signals.
6. Initialize the DMA module clock.
7. Configure DMA_CHSPERC, DMA_CHCTRLC, and REG_DMA_CHDPERC to set the source and

destination data transfer formats and select the transfer channel.

8. Configure DMA_SRCADDRC and REG_DMA_CHCTRLC to set the source and destination addresses and set the transfer block size.
9. Set DMA_EN = 1 to enable DMA transfer.
10. Enable LPTIMER_CR1.EN to start the counter.

15.7.3 PWM Output

1. Initialize the LPTIMER clock module.
2. Configure the pin to be multiplexed as LPTIMER_OUT.
3. Configure LPTIMER_CR2.SINGLE to set the counting mode.
4. Configure LPTIMER_CR2.DIVCFG to set the division factor.
5. Configure LPTIMER_CR2.CLKSEL to set the clock source.
6. Configure the value of the LPTIMER_CCR1 capture/compare register.
7. Configure the value of the LPTIMER_ARR1 target register.
8. Configure LPTIMER_CCMCFG1.CC1P to select the PWM output waveform polarity.
9. Configure LPTIMER_CCMCFG1.CC1S to select the PWM output mode.
10. Configure LPTIMER_CCMCFG1.CC1E to enable the LPTIMER0 capture/compare function.
11. Configure the LPTIMER_IER interrupt register to enable interrupts.
12. Enable LPTIMER_CR1.EN to start the counter.

15.7.4 ETR Pulse Trigger Counting Mode

1. Initialize the LPTIMER clock module.

2. Configure the pin to be multiplexed as LPTIMER_ETR.
3. Configure LPTIMER_CR2.SINGLE to set the counting mode.
4. Configure LPTIMER_CR2.DIVCFG to set the division factor.
5. Configure LPTIMER_CR2.CLKSEL to set the clock source for sampling the ETR signal.
6. Configure the value of the LPTIMER_ARR1 target register.
7. Configure LPTIMER_CR2.TRIGEDGE to set the ETR trigger edge.
8. Configure LPTIMER_CR2.LPMOD to select the pulse trigger counting mode.
9. Configure the LPTIMER_IER.TIE interrupt register to enable the external trigger interrupt.
10. Enable LPTIMER_CR1.EN to start the counter.

15.7.5 ETR Pulse Counting Mode

1. Initialize the LPTIMER clock module.
2. Configure the pin to be multiplexed as LPTIMER_IN.
3. Configure LPTIMER_CR2.SINGLE to set the counting mode.
4. Configure LPTIMER_CR2.DIVCFG to set the division factor.
5. Configure LPTIMER_CR2.CLKSEL to set the clock source to 0x011, using the LPTIMER_IN input as the counting clock.
6. Configure the value of the LPTIMER_ARR1 target register.
7. Configure LPTIMER_CR2.EDGESEL to set the ETR clock input edge.
8. Configure LPTIMER_CR2.LPMOD to select the ETR pulse counting mode.
9. Configure the LPTIMER_IER.TIE interrupt register to enable interrupts.
10. Enable LPTIMER_CR1.EN to start the counter.

15.7.6 Timeout Mode

1. Initialize the LPTIMER clock module.
2. Configure the pin to be multiplexed as LPTIMER_EXT.
3. Configure LPTIMER_CR2.SINGLE to set the counting mode.
4. Configure LPTIMER_CR2.DIVCFG to set the division factor.
5. Configure LPTIMER_CR2.CLKSEL to set the clock source.
6. Configure the value of the LPTIMER_ARR1 target register.
7. Configure LPTIMER_CR2.TRIGEDGE to set the external trigger edge.
8. Configure LPTIMER_CR2.LPMOD to select the timeout mode.
9. Configure the LPTIMER_IER interrupt register to enable the overflow interrupt.
10. Enable LPTIMER_CR1.EN to start the counter.

Note: If no new trigger occurs before the counter overflows, an overflow interrupt will be generated, the counting will be stopped, and the enable will be cleared. To reuse the counter, the interrupt shall be enabled again.

16 DMA

16.1 Overview

Direct memory access (DMA) supports 2-channel data transfer.

16.2 Main Features

- Single master port
- Can control data transfer between FLASH, SRAM, SPI, I2C, ADC, LPTIMER, GTIMER and ATIMER modules, where FLASH can only serves as the source address
- Supporting memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral data transfers
- Internally provided with 2 DMA channels
- Configurable bit width and block length of data transfer
- Supporting fixed, incremental and decremental transfers for both source address and destination address, and supporting infinite transfer
- Burst transfer supported with configurable transfer counts ranging from 2 to 16

16.3 Register Description

DMA register base address: 0x40020000

Offset	Name	Description
0x00	DMA_SRCADDR_C0	DMA channel 0 source transfer address register
0x04	DMA_DSTADDR_C0	DMA channel 0 destination transfer address register
0x08	DMA_CHCTRL_C0	DMA channel 0 control information register

Offset	Name	Description
0x0C	DMA_CHSTS_C0	DMA channel 0 transfer status register
0x10	DMA_CHSPER_C0	DMA channel 0 source peripheral selection register
0x14	DMA_CHDPER_C0	DMA channel 0 destination peripheral selection register
0x20	DMA_SRCADDR_C1	DMA channel 1 source transfer address register
0x24	DMA_DSTADDR_C1	DMA channel 1 destination transfer address register
0x28	DMA_CHCTRL_C1	DMA channel 1 control information register
0x2C	DMA_CHSTS_C1	DMA channel 1 transfer status register
0x30	DMA_CHSPER_C1	DMA channel 1 source peripheral selection register
0x34	DMA_CHDPER_C1	DMA channel 1 destination peripheral selection register
0x40	DMA_EN	DMA controller enable register
0x44	DMA_SOFT_RESET	DMA soft reset register
0x48	DMA_INT_STATUS	DMA interrupt status register
0x4C	DMA_INT_MASK	DMA interrupt mask register
0x54	DMA_PER_REQ	DMA peripheral request status register

16.3.1 DMA Channel Source Transfer Address Register (DMA_SRCADDR_Cx) (Offset: 20 * x + 00H) (x = 0, 1)

Bit	Name	Attribute	Reset Value	Description
31:0	SRC_ADDR	RW	0x0	Source address

16.3.2 DMA Channel Destination Transfer Address Register (DMA_DSTADDR_Cx) (Offset: 20 * x + 04H) (x = 0, 1)

Bit	Name	Attribute	Reset Value	Description
31:0	DST_ADDR	RW	0x0	Destination address

16.3.3 DMA Channel Control Information Register (DMA_CHCTRL_Cx) (Offset: 20 * x + 08H) (x = 0, 1)

<table>
<tr><th>Bit</th><th>Name</th><th>Attribute</th><th>Reset Value</th><th>Description</th></tr>
<tr><td>31:30</td><td>WIDTH</td><td>RW</td><td>0x0</td><td>Data bit width:
0: 8-bit data width
1: 16-bit data width
2: 32-bit data width
3: Invalid, but the module behaves as if it is performing 32-bit read/write operations with the same bit widths for source and destination data</td></tr>
<tr><td>29:15</td><td>XFER_SIZE</td><td>RW</td><td>0x0</td><td>Transfer block size:
Supporting blocks of 32767 bytes for 8-bit data width
Supporting blocks of 32767 halfwords for 16-bit data width
Supporting blocks of 32767 words for 32-bit data width
Note: The block size is cleared after the current block transfer is completed.</td></tr>
<tr><td>14:12</td><td>FLOW_CTRL</td><td>RW</td><td>0x0</td><td>Flow control mode:
<table>
<tr><th>Data Value</th><th>Source</th><th>Destination</th><th>Flow Control</th></tr>
<tr><td>0</td><td>Memory</td><td>Memory</td><td>DMA</td></tr>
<tr><td>1</td><td>Memory</td><td>Peripheral</td><td>DMA</td></tr>
<tr><td>2</td><td>Peripheral</td><td>Memory</td><td>DMA</td></tr>
<tr><td>3</td><td>Peripheral</td><td>Peripheral</td><td>DMA</td></tr>
</table></td></tr>
<tr><td>11</td><td>TRANS_FREE</td><td>RW</td><td>0x0</td><td>Infinite transfer enable:
1: Enable infinite transfer
0: Disable infinite transfer
The infinite transfer means the transfer quantity is not controlled by the XFER_SIZE register and can continue indefinitely.</td></tr>
<tr><td>10</td><td>WAP_SRC_EN</td><td>RW</td><td>0x0</td><td>SRC burst function enable:
1: Enable SRC burst function</td></tr>
</table>

Bit	Name	Attribute	Reset Value	Description
				0: Disable SRC burst function The SRC burst function enables burst transfer, which can cyclically read data from 2, 3, 4, 5, 6, 7, 8, or 9 consecutive address spaces. The reading direction can be either incremental or decremental.
9:8	RSV	-	-	Reserved
7	WAP_DST_EN	RW	0x0	DST burst function enable: 1: Enable DST burst function 0: Disable DST burst function The DST burst function enables burst transfer, which can cyclically read data from 2, 3, 4, 5, 6, 7, 8, or 9 consecutive address spaces. The reading direction can be either incremental or decremental.
6:5	RSV	-	-	Reserved
4:3	DST_INC	RW	0x0	Indication bit of destination address increment: if enabled, the destination address will increment with the read data, otherwise it remains unchanged. 01: Address increment 10: Address decrement
2:1	SRC_INC	RW	0x0	Indication bit of source address increment: if enabled, the source address will increment with the read data, otherwise it remains unchanged. 01: Address increment 10: Address decrement
0	CH_EN	RW	0x0	Channel enable flag, which will be automatically cleared upon completion of block transfer for DMA flow control.

16.3.4 DMA Channel Transfer Status Register (DMA_CHSTS_Cx) (Offset: 20 * x + 0CH) (x = 0, 1)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:1	LENGTH	R	0x0	During DMA transfer, it indicates the length of data that has been transferred via this channel.
0	CH_BUSY	R	0x0	Channel status: 0: Idle 1: Busy

16.3.5 DMA Channel Source Peripheral Selection Register (DMA_CHSPER_Cx) (Offset: 20 * x + 10H) (x = 0, 1)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:5	WAP_SRC_NUM	RW	0x0	SRC burst transfer count configuration: 0: SRC burst 2 1: SRC burst 3 2: SRC burst 4 3: SRC burst 5 4: SRC burst 6 5: SRC burst 7 6: SRC burst 8 7: SRC burst 9
4:0	SPER	RW	0x0	Source peripherals, mainly used for the request selection of source peripherals, with the specific peripheral assignments as follows: 1: SPI receive 2: I2C receive 4: ADC receive 5: GTIMER0_UEV

Bit	Name	Attribute	Reset Value	Description
				6: GTIMER0_TG 7: GTIMER0_CC4 8: GTIMER0_CC3 9: GTIMER0_CC2 10: GTIMER0_CC1 11: GTIMER1_UEV 12: GTIMER1_TG 13: GTIMER1_CC4 14: GTIMER1_CC3 15: GTIMER1_CC2 16: GTIMER1_CC1 17: GTIMER2_UEV 18: GTIMER2_TG 19: GTIMER2_CC4 20: GTIMER2_CC3 21: GTIMER2_CC2 22: GTIMER2_CC1 23: LPTIMER_RX2 24: LPTIMER_RX1 25: ATIMER_CTU 26: ATIMER_CC4 27: ATIMER_CC3 28: ATIMER_CC2 29: ATIMER_CC1 30: ATIMER_TG 31: ATIMER_UEV

16.3.6 DMA Channel Destination Peripheral Selection Register (DMA_CHDPER_Cx) (Offset: 20 * x + 14H) (x = 0, 1)

Bit	Name	Attribute	Reset Value	Description
31:8	RSV	-	-	Reserved
7:5	WAP_DST_NUM	RW	0x0	DST burst transfer count configuration: 0: DST burst 2 1: DST burst 3

Bit	Name	Attribute	Reset Value	Description
				2: DST burst 4 3: DST burst 5 4: DST burst 6 5: DST burst 7 6: DST burst 8 7: DST burst 9
4:0	DPER	RW	0x0	Destination peripherals, mainly used for the request selection of destination peripherals, with the specific peripheral allocations as follows: 0: SPI transmit 3: I2C transmit 5: GTIMER0_UEV 6: GTIMER0_TG 7: GTIMER0_CC4 8: GTIMER0_CC3 9: GTIMER0_CC2 10: GTIMER0_CC1 11: GTIMER1_UEV 12: GTIMER1_TG 13: GTIMER1_CC4 14: GTIMER1_CC3 15: GTIMER1_CC2 16: GTIMER1_CC1 17: GTIMER2_UEV 18: GTIMER2_TG 19: GTIMER2_CC4 20: GTIMER2_CC3 21: GTIMER2_CC2 22: GTIMER2_CC1 25: ATIMER_CTU 26: ATIMER_CC4 27: ATIMER_CC3 28: ATIMER_CC2 29: ATIMER_CC1 30: ATIMER_TG 31: ATIMER_UEV

16.3.7 DMA Controller Enable Register (DMA_EN) (Offset: 40H)

Bit	Name	Attribute	Reset Value	Description
31:2	RSV	-	-	Reserved
1	GRAND_SET	RW	0x0	1: Enable the polling function of channel priority, the channel priority will be polled in the next transfer (initially, channel 1 has higher priority than channel 0; if channel 1 has higher priority than channel 0 in the current transfer, channel 0 will have higher priority than channel 1 in the next transfer). 0: Fixed priority between channels (by default, channel 0 has higher priority than channel 1)
0	DMA_EN	RW	0x0	1: Enable DMA controller 0: Disable DMA controller

16.3.8 DMA Soft Reset Register (DMA_SOFT_RESET) (Offset: 44H)

Bit	Name	Attribute	Reset Value	Description
31:0	DMA_SOFT_RESET	W	0x0	This register is a virtual register for write operation. When the DMA module samples a write operation to this register with any value, DMA will reset the state machine and the register to be reset.

16.3.9 DMA Interrupt Status Register (DMA_INT_STATUS) (Offset: 48H)

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3	INT_TC_C1	W1C	0x0	Channel 1 block transfer completion interrupt indicator, can be cleared by writing 1

Bit	Name	Attribute	Reset Value	Description
2	INT_TC_C0	W1C	0x0	Channel 0 block transfer completion interrupt indicator, can be cleared by writing 1
1	INT_ERR_C1	W1C	0x0	Channel 1 bus error interrupt indicator, can be cleared by writing 1
0	INT_ERR_C01	W1C	0x0	Channel 0 bus error interrupt indicator, can be cleared by writing 1

16.3.10 DMA Interrupt Mask Register (DMA_INT_MASK) (Offset: 4CH)

Bit	Name	Attribute	Reset Value	Description
31:4	RSV	-	-	Reserved
3	MASK_TC_C1	RW	0x0	Channel 1 block transfer completion interrupt mask: if it is low, IntTc interrupt will not be output, i.e., IntTc = 0.
2	MASK_TC_C0	RW	0x0	Channel 0 block transfer completion interrupt mask: if it is low, IntTc interrupt will not be output, i.e., IntTc = 0.
1	MASK_ERR_C1	RW	0x0	Channel 1 bus error interrupt mask: if it is low, IntErr interrupt will not be output, i.e., IntErr = 0.
0	MASK_ERR_C0	RW	0x0	Channel 0 bus error interrupt mask: if it is low, IntErr interrupt will not be output, i.e., IntErr = 0.

16.3.11 DMA Peripheral Request Register (DMA_PER_REQ) (Offset: 54H)

Bit	Name	Attribute	Reset Value	Description
31	ATIMER_UEV_REQ	R	0x0	ATIMER update event request
30	ATIMER_TG_REQ	R	0x0	DMA external trigger request
29	ATIMER_CC1_REQ	R	0x0	ATIMER channel 1 request
28	ATIMER_CC2_REQ	R	0x0	ATIMER channel 2 request
27	ATIMER_CC3_REQ	R	0x0	ATIMER channel 3 request
26	ATIMER_CC4_REQ	R	0x0	ATIMER channel 4 request

Bit	Name	Attribute	Reset Value	Description
25	ATIMER_CTU_REQ	R	0x0	ATIMER COM event request
24	LPTIMER_RX0_REQ	R	0x0	LPTIMER RX0 receive request
23	LPTIMER_RX1_REQ	R	0x0	LPTIMER RX1 receive request
22	GTIMER2_CC1_REQ	R	0x0	GTIMER2 channel 1 request
21	GTIMER2_CC2_REQ	R	0x0	GTIMER2 channel 2 request
20	GTIMER2_CC3_REQ	R	0x0	GTIMER2 channel 3 request
19	GTIMER2_CC4_REQ	R	0x0	GTIMER2 channel 4 request
18	GTIMER2_TG_REQ	R	0x0	GTIMER2 external trigger request
17	GTIMER2_UEV_REQ	R	0x0	GTIMER2 update event request
16	GTIMER1_CC1_REQ	R	0x0	GTIMER1 channel 1 request
15	GTIMER1_CC2_REQ	R	0x0	GTIMER1 channel 2 request
14	GTIMER1_CC3_REQ	R	0x0	GTIMER1 channel 3 request
13	GTIMER1_CC4_REQ	R	0x0	GTIMER1 channel 4 request
12	GTIMER1_TG_REQ	R	0x0	GTIMER1 external trigger request
11	GTIMER1_UEV_REQ	R	0x0	GTIMER1 update event request
10	GTIMER0_CC1_REQ	R	0x0	GTIMER0 channel 1 request
9	GTIMER0_CC2_REQ	R	0x0	GTIMER0 channel 2 request
8	GTIMER0_CC3_REQ	R	0x0	GTIMER0 channel 3 request
7	GTIMER0_CC4_REQ	R	0x0	GTIMER0 channel 4 request
6	GTIMER0_TG_REQ	R	0x0	GTIMER0 external trigger request
5	GTIMER0_UEV_REQ	R	0x0	GTIMER0 update event request
4	ADC_REQ	R	0x0	ADC receive request
3	I2C_TX_REQ	R	0x0	I2C transmit request
2	I2C_RX_REQ	R	0x0	I2C receive request
1	SPI_RX_REQ	R	0x0	SPI receive request
0	SPI_TX_REQ	R	0x0	SPI transmit request

16.4 Operation Procedure

Software configuration steps:

1. Configure DMA_CHCTRL_Cx[14:12] to select the DMA transfer mode.
2. Configure DMA_CHCTRL_Cx[4:0] and DMA_CHDPER_Cx[4:0] to select the peripheral handshake signal (only required when the transfer address is a peripheral).

3. Configure DMA_CHCTRL_Cx[4:3] and DMA_CHCTRL_Cx[2:1] to select whether the source address and destination address will increment or remain unchanged.
4. Configure DMA_CHCTRL_Cx[31:30] to select the bit width of the transferred data.
5. Set DMAC_EN[0] to 1 to enable the DMA controller.
6. Configure DMA_SRCADDR_Cx to set the channel source address.
7. Configure DMA_DSTADDR_Cx to set the channel destination address.
8. Configure DMA_CHCTRL_Cx[29:15] to set the number of transfer blocks.
9. Set DMA_CHCTRL_Cx[0] to 1 to enable DMA channel transfer.
10. Wait for DMA_CHCTRL_Cx[0] to be 0, indicating the transfer is complete. If the transfer completion interrupt is enabled, wait for the interrupt before proceeding.

17 CRC

17.1 Overview

CRC is a hardware 16-bit CRC cyclic redundancy check calculation circuit with the polynomial $G(x) = x^{16} + x^{12} + x^5 + 1$. It can calculate the appropriate CRC result based on the user-preset CRC initial value and communication data, and supports setting the direction (forward and reverse) of input data and results.

17.2 Register Description

CRC register base address: 0x4000_1800

Table 17-1: List of CRC Registers

Offset	Name	Description
0x00	CRC_DATA	CRC data register
0x04	CRC_INIT	CRC initial value register
0x08	CRC_CTRL	CRC control register

17.2.1 Data Register CRC_DATA (Offset: 00H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:8	RSLT2	R	0x0	Read the upper 8 bits of the 16-bit CRC calculation result.
7:0	RSLT1	RW	0x0	Write: Write the data that requires CRC check calculation. If the data to be checked is more than 8 bits, it shall be written sequentially. Read: Read the lower 8 bits of the 16-bit CRC calculation result.

The lower 8 bits are write-only for the data to be checked, and cannot be read again after writing.

A read operation returns a 16-bit CRC calculation result, of which the lower 8 bits are multiplexed with the data register.

A read operation will reset the CRC calculation, meaning that after the read, the initial value will be reloaded, and the next input data will start a new round of calculation independent of the previous result.

17.2.2 Initial Value Register CRC_INIT (Offset: 04H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:0	INIT	RW	0x0	Write the 16-bit CRC initial value.

17.2.3 Control Register CRC_CTRL (Offset: 08H)

Bit	Name	Attribute	Reset Value	Description
31:3	RSV	-	-	Reserved
2	RSLT_REV	RW	0x0	CRC calculation result bit-reversal: 1: Reversed 0: Not reversed
1	DATA_REV	RW	0x0	CRC calculation data bit-reversal: 1: Reversed 0: Not reversed
0	INITIAL_REV	RW	0x0	CRC initial value bit-reversal: 1: Reversed 0: Not reversed

17.3 Operation Procedure

1. Set the 16-bit initial value CRC_INIT[15:0].
2. Set CRC_CTRL[2:0] to select whether the data is reversed.
3. Write the 8-bit CRC calculation data into CRC_DATA. If the CRC data input is not completed,

sequentially input the subsequent 8 bits of data until all data is completed.entered.

4. Read CRC_DATA, which will return the CRC calculation result once.

Note: After reading the result, the CRC calculation module will end the current calculation and reload the initial value for the next calculation.

18 RNG

18.1 Overview

RNG is a random number generator that can generate different random number sequences by writing different random number seeds.

18.2 Main Features

- 32-bit random numbers
- Continuous reading of random number sequences

18.3 Register Description

Register base address: 0x4000_2000

Table 18-1: List of RNG Registers

Offset	Name	Description
0x0E0	RNG_CR	Random number control register
0x0E4	RNG_SEED	Random number seed register
0x0E8	RNG_DATA	Random number data register

18.3.1 Random Number Control Register RNG_CR (Offset: 0E0H)

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	R	0	Reserved
0	EN	RW	0x0	1: Enable random number generator 0: Disable random number generator

18.3.2 Random Number Seed Register RNG_SEED (Offset: 0E4H)

Bit	Name	Attribute	Reset Value	Description
31:0	SEED	RW	0x0	Random number seed register

18.3.3 Random Number Data Register (Offset: 0E8H)

Bit	Name	Attribute	Reset Value	Description
31:0	DATA	R	0xFF00FF	Random number data register, from which the random number value can be read out.

18.4 Operation Procedure

1. Set RNG_CR[0] to 1 to enable the random number generator.
2. Write the random number seed value to RNG_SEED[31:0].
3. Read RNG_DATA[31:0] to obtain the random number value, which can be read continuously.

19 WDT

19.1 Overview

The watchdog timer can generate a non-maskable interrupt or a reset when it reaches the timeout value. It can be used to regain control when the system fails to respond as expected due to software errors or external device failures.

19.2 Main Features

- 32-bit downcounter with programmable load
- Independent watchdog timer enable
- Interrupt generation logic with interrupt masking
- Software runaway protection lock register
- Reset enable/disable generation logic
- User-enabled hault when the CPU of the microprocessor is suspended during debugging

19.3 Register Description

WDT register base address: 0x40002400

Table 19-1: List of WDT Registers

Offset	Name	Description
0x00	WDT_LOAD	Load register
0x04	WDT_CNT	Counter register
0x08	WDT_CTRL	Control register
0x0C	WDT_CLR	Clear register
0x10	WDT_INTRAW	RAW interrupt status register

Offset	Name	Description
0x14	WDT_MINTS	MASK interrupt status register
0x18	WDT_STALL	Stall register
0x1C	WDT_LOCK	Lock register

19.3.1 Load Register WDT_LOAD (Offset: 00H)

Bit	Name	Attribute	Default Value	Functional Description
31:0	LOAD	RW	0xFFFFFFFF	WDOG initial load value

19.3.2 Counter Register WDT_CNT (Offset: 04H)

Bit	Name	Attribute	Default Value	Functional Description
31:0	CNT	R	0xFFFFFFFF	Counter value in WDOG

19.3.3 Control Register WDT_CTRL (Offset: 08H)

Bit	Name	Attribute	Default Value	Functional Description
31	WRC	R	0x1	Set the WDT load value or write to the WDT_CTRL register to make this bit effective. When writing to the WDT_LOAD or WDT_CTRL register, there will be a certain delay before the setting bit takes effect. 0: The setting has not yet taken effect. 1: The setting has taken effect.
30:2	RSV	-	-	Reserved
1	RSTEN	RW	0x0	WDT overflow reset enable: 0: Disabled 1: Enabled
0	INTEN	RW	0x0	WDT interrupt enable: 0: Disabled 1: Enabled

19.3.4 Clear Register WDT_CLR (Offset: 0CH)

Bit	Name	Attribute	Default Value	Functional Description
31:0	CLR_CARRY	W	0x0	Writing any value to this register will clear the WDT overflow status, thus clearing the interrupt and reset.

19.3.5 RAW Interrupt Status Register WDT_INTRAW (Offset: 10H)

Bit	Name	Attribute	Default Value	Functional Description
31:0	INTRAW	R	0x0	Raw interrupt register, unmasked by interrupt enable: 0: No overflow has occurred inside the WDT. 1: An overflow has occurred inside the WDT.

19.3.6 Mask Interrupt Status Register WDT_MINTS (Offset: 14H)

Bit	Name	Attribute	Default Value	Functional Description
31:0	INTMS	R	0x0	0: No interrupt has occurred in WDT. 1: An interrupt has occurred in WDT.

19.3.7 STALL Control Register WDT_STALL (Offset: 18H)

Bit	Name	Attribute	Default Value	Functional Description
31:9	RSV	-	-	Reserved
8	STALL	RW	0x0	Enable bit for WDT not counting when the chip is in HALT state: 0: Disable the function that stops the counter in HALT state. 1: Enable the function that stops the counter in HALT state.
7:0	RSV	-	-	Reserved

19.3.8 LOCK Register WDT_LOCK (Offset: 1CH)

Bit	Name	Attribute	Default Value	Functional Description
31:0	LOCK	W	0x0	WDT LOCK function enable: When the LOCK function is enabled, all WDT registers except this one are not writable. Writing any value to this register enables the WDT LOCK function, while writing 0x1ACCE551 to this register clears the LOCK function.

19.4 Operation Procedure

19.4.1 WDT Timer Configuration

1. Write 0x1ACCE551 to the WDT_LOCK register to unlock the register.
2. Set the prescaler value in the WDT_STALL register.
3. Enable the INTEN interrupt function in the WDT_CTRL register.
4. Wait for the WRC bit in the WDT_CTRL register to be set, indicating that the setting has taken effect.
5. Load the desired value into the WDT_LOAD register.
6. Wait for the WRC bit in the WDT_CTRL register to be set, indicating that the setting has taken effect.
7. Write any value to the WDT_LOCK register to lock the register.
8. The system enters the interrupt periodically according to the loaded value and the prescaler value of the counter clock.

19.4.2 WDT Watchdog Feeding Configuration

1. Write 0x1ACCE551 to the WDT_LOCK register to unlock the register.
2. Set the prescaler value in the WDT_STALL register.
3. Enable the RSTEN reset function in the WDT_CTRL register, and optionally enable the INTEN interrupt function; wait until the WRC bit of the WDT_CTRL register is set, indicating that the setting has taken effect.
4. Load the desired value into the WDT_LOAD register.
5. Wait for the WRC bit in the WDT_CTRL register to be set, indicating that the setting has taken effect.
6. Write any value to the WDT_LOCK register to lock the register.
7. The system can feed the watchdog periodically by updating the WDT_LOAD through software or clearing the interrupt flag bit. If the dog is not fed within the timing period, the system will reset after twice the timing period.

20 WWDT

20.1 Overview

The window watchdog timer (WWDT) is a watchdog running synchronously with CPU, aiming at monitoring the operating status of CPU in real time, so that it can reset CPU in the case of abnormal operation to avoid unpredictable consequences.

20.2 Main Features

- 18-bit upcounter with programmable load
- Internal fault detector of the system
- Clock synchronized with the system clock
- Used for monitoring software errors
- Reset will be triggered if the watchdog is fed before the window or not fed within the timeout period (the valid window for feeding the watchdog is 50%-100% of the time)
- An early warning interrupt will be triggered when the counter reaches 75% of the overflow time.

20.3 Register Description

WWDT register base address: 0x40003C00

Table 20-1: List of WWDT Registers

Offset	Name	Description
0x00	WWDT_CON	Control register
0x04	WWDT_CFG	Configuration register

Offset	Name	Description
0x08	WWDT_CNT	Counter register
0x0C	WWDT_IE	Interrupt enable register
0x10	WWDT_IF	Interrupt flag register

20.3.1 Control Register WWDT_CON (Offset: 00H)

Bit	Name	Attribute	Default Value	Functional Description
31:8	RSV	-	-	Reserved
7:0	CON	W	0x0	CPU writing 0x5A to this address will start the WWDT timer. After starting the WWDT, CPU writing 0xAC to this address will clear the counter.

20.3.2 Configuration Register WWDT_CFG (Offset: 04H)

Bit	Name	Attribute	Default Value	Functional Description
31:5	RSV	-	-	Reserved
4	STOP_EN	RW	0x1	WWDT break enable bit: after software sets a breakpoint, the counter is prohibited from counting: 0: Counter operates normally. 1: Counter stops counting.
3:0	CFG	RW	0x0	Configure watchdog overflow time: 0000: t_{PCLK}* 4096 * 1 0001: t_{PCLK}* 4096 * 4 0010: t_{PCLK}* 4096 * 16 0011: t_{PCLK}* 4096 * 64 0100: t_{PCLK}* 4096 * 128 0101: t_{PCLK}* 4096 * 256 0110: t_{PCLK}* 4096 * 512 0111: t_{PCLK}* 4096 * 1024 1000: t_{PCLK}* 4096 * 2048 1001: t_{PCLK}* 4096 * 4096 1010: t_{PCLK}* 4096 * 8192 1011: t_{PCLK}* 4096 * 16384 1100: t_{PCLK}* 4096 * 32768

Bit	Name	Attribute	Default Value	Functional Description
				1101: t_{PCLK}* 4096 * 65536 1110: t_{PCLK}* 4096 * 131072 1111: t_{PCLK}* 4096 * 262144

20.3.3 Counter Register WWDT_CNT (Offset: 08H)

Bit	Name	Attribute	Default Value	Functional Description
31:18	RSV	-	-	Reserved
17:0	CNT	R	0x0	WWDT counter register value. Software can query this register to know the WWDT timing progress.

20.3.4 Interrupt Enable Register WWDT_IE (Offset: 0CH)

Bit	Name	Attribute	Default Value	Functional Description
31:1	RSV	-	-	Reserved
0	IE	RW	0x0	WWDT interrupt enable: 0: Disabled 1: Enabled

20.3.5 Interrupt Flag Register WWDT_IF (Offset: 10H)

Bit	Name	Attribute	Default Value	Functional Description
31:1	RSV	-	-	Reserved
0	IF	W1C	0x0	75% timing interrupt flag of WWDT, can be cleared by writing 1: 0: No interrupt generated 1: Interrupt flag set

20.4 Operation Procedure

- WWDT timer configuration:

  1. Set the overflow time length in the WWDT_CFG register.
  2. Enable the interrupt in the WWDT_IE register.
  3. Write 0x5A to the WWDT_CON register to start the WWDT timer.
  4. Wait for the interrupt to occur (interrupt generated at 75% of the timing).
  5. Wait for the reset to occur (reset generated after overflow).

- WWDT dog feeding configuration:

  Write 0xAC to the WWDT_CON register to clear the counter within 50%–100% of the counting time.

21 ADC

21.1 Overview

This is a 12-bit successive approximation analog-to-digital converter (ADC). It features up to 16 input channels, capable of measuring signals from 11 external sources, 3 internal PGA outputs, 1 internal LDO output, and 1 internal 1/4 VDDH output. The A/D conversion of these channels can be performed in either single-shot or continuous scan mode. The ADC controller facilitates the communication between CPU and SAR ADC. The result of ADC conversion is stored in the lower 12 bits of the data register.

21.2 Main Features

- Supporting DMA transfer mode
- 5-bit programmable divider for generating the A/D clock
- 12-bit resolution A/D input data, with a maximum sampling rate of 1 MSPS (configurable via software)
- 16-channel ADC input: 11 pin channels, 3 internal PGA outputs, 1 internal LDO output, and 1 internal 1/4 VDDH output
- Analog ADC can be disabled
- Polling and interrupt transfer modes
- Single-shot scan or continuous scan mode
- Interrupt sources: single conversion complete interrupt, conversion sequence complete interrupt, injected channel single conversion complete interrupt, injected channel

conversion sequence complete interrupt, analog watchdog upper limit exceed interrupt, analog watchdog lower limit drop interrupt, data conflict interrupt; channel data valid flags (one flag for each of the 8 regular channels, 4 injected channels, and oversampling)

- Supporting on-chip peripheral-triggered ADC conversion
- ADC voltage input range: 0–V_{ref}
- Optional ADC reference voltage source: chip supply voltage V_{DDH}, external voltage on IO pin V_{REFIO}, and internal V_{REF}

21.3 ADC Pin Description

Table 21-1: ADC Pin Description

Channel	Pin Name
ADC_CH0	AIN[0], PC2
ADC_CH1	AIN[1], PD0
ADC_CH2	AIN[2], PD1
ADC_CH3	AIN[3], PD2
ADC_CH4	AIN[4], PD3
ADC_CH5	AIN[5], PB7
ADC_CH6	AIN[6], PB4
ADC_CH7	AIN[7], PB3
ADC_CH8	AIN[8], PB2
ADC_CH9	AIN[9], PB1
ADC_CH10	AIN[10], PB0
ADC_CH11	AIN[11], PGA0_VOUT
ADC_CH12	AIN[12], PGA1_VOUT
ADC_CH13	AIN[13], PGA2_VOUT
ADC_CH14	Internal 1/4 V_{DDH}
ADC_CH15	Internal V_{REF}

21.4 Register Description

ADC register base address: 0x4000_1C00

Table 21-2: List of ADC Registers

Offset	Name	Description
0x00	ADC_ISR	ADC interrupt and status register
0x04	ADC_IER	ADC interrupt enable register
0x08	ADC_CR	ADC control register
0x0C	ADC_CFGR	ADC configuration register
0x10	ADC_SMTR	ADC sampling time control register
0x14	ADC_CHER	ADC channel control register
0x18	ADC_SEQ	ADC channel selection register
0x20	ADC_HLTR	ADC analog watchdog threshold register
0x24	ADC_IJCHER	ADC injected channel control register
0x28	ADC_IJSEQ	ADC injected channel selection register
0x30	ADC_DR0	ADC data register 0
0x34	ADC_DR1	ADC data register 1
0x38	ADC_DR2	ADC data register 2
0x3C	ADC_DR3	ADC data register 3
0x40	ADC_DR4	ADC data register 4
0x44	ADC_DR5	ADC data register 5
0x48	ADC_DR6	ADC data register 6
0x4C	ADC_DR7	ADC data register 7
0x50	ADC_IJDR0	ADC injected channel data register 0
0x54	ADC_IJDR1	ADC injected channel data register 1
0x58	ADC_IJDR2	ADC injected channel data register 2
0x5C	ADC_IJDR3	ADC injected channel data register 3
0x60	ADC_DR_OVS	ADC oversampling average result register
0xA8	ADC_HDT	ADC hardware trigger enable register
0xAC	ADC_IJHDT	ADC injected channel hardware trigger enable register
0xB0	ADC_SMTR0	ADC channel sampling time configuration register 0
0xB4	ADC_SMTR1	ADC channel sampling time configuration register 1
0xB8	ADC_IJSMTR	ADC injected channel sampling time configuration register

21.4.1 ADC Interrupt and Status Register (ADC_ISR) (Offset: 000H)

Bit	Name	Attribute	Reset Value	Description
31:29	RSV	-	-	Reserved
28	DR_OVS_V	W1C	0x0	Data valid flag of the ADC_DR_OVS register, cleared by reading the ADC_DR_OVS register or writing 1: 0: No new data in the ADC_DR_OVS register. 1: New data present in the ADC_DR_OVS register.
27	IJDR3_V	W1C	0x0	Data valid flag of the ADC_IJDR3 register, cleared by reading the ADC_IJDR3 register or writing 1: 0: No new data in the ADC_IJDR3 register. 1: New data present in the ADC_IJDR3 register.
26	IJDR2_V	W1C	0x0	Data valid flag of the ADC_IJDR2 register, cleared by reading the ADC_IJDR2 register or writing 1: 0: No new data in the ADC_IJDR2 register. 1: New data present in the ADC_IJDR2 register.
25	IJDR1_V	W1C	0x0	Data valid flag of the ADC_IJDR1 register, cleared by reading the ADC_IJDR1 register or writing 1: 0: No new data in the ADC_IJDR1 register. 1: New data present in the ADC_IJDR1 register.
24	IJDR0_V	W1C	0x0	Data valid flag of the ADC_IJDR0 register, cleared by reading the ADC_IJDR0 register or writing 1: 0: No new data in the ADC_IJDR0 register. 1: New data present in the ADC_IJDR0 register.
23	DR7_V	W1C	0x0	Data valid flag of the ADC_DR7 register, cleared by reading the ADC_DR7 register or writing 1: 0: No new data in the ADC_DR7 register. 1: New data present in the ADC_DR7 register.
22	DR6_V	W1C	0x0	Data valid flag of the ADC_DR6 register, cleared by reading the ADC_DR6 register or writing 1: 0: No new data in the ADC_DR6 register. 1: New data present in the ADC_DR6 register.
21	DR5_V	W1C	0x0	Data valid flag of the ADC_DR5 register, cleared by reading the ADC_DR5 register or writing 1: 0: No new data in the ADC_DR5 register. 1: New data present in the ADC_DR5 register.

Bit	Name	Attribute	Reset Value	Description
20	DR4_V	W1C	0x0	Data valid flag of the ADC_DR4 register, cleared by reading the ADC_DR4 register or writing 1: 0: No new data in the ADC_DR4 register. 1: New data present in the ADC_DR4 register.
19	DR3_V	W1C	0x0	Data valid flag of the ADC_DR3 register, cleared by reading the ADC_DR3 register or writing 1: 0: No new data in the ADC_DR3 register. 1: New data present in the ADC_DR3 register.
18	DR2_V	W1C	0x0	Data valid flag of the ADC_DR2 register, cleared by reading the ADC_DR2 register or writing 1: 0: No new data in the ADC_DR2 register. 1: New data present in the ADC_DR2 register.
17	DR1_V	W1C	0x0	Data valid flag of the ADC_DR1 register, cleared by reading the ADC_DR1 register or writing 1: 0: No new data in the ADC_DR1 register. 1: New data present in the ADC_DR1 register.
16	DR0_V	W1C	0x0	Data valid flag of the ADC_DR0 register, cleared by reading the ADC_DR0 register or writing 1: 0: No new data in the ADC_DR0 register. 1: New data present in the ADC_DR0 register.
15:9	RSV	-	-	Reserved
8	IJ_EOS	W1C	0x0	Injected channel conversion sequence end flag (IJ_EOS): set when conversion of all enabled channels is completed; cleared by software writing 1.
7	IJ_EOC	W1C	0x0	Injected channel single-shot conversion end flag (IJ_EOC): set after the conversion of each channel is completed; cleared by software writing 1 or reading the ADC_IJDR0–ADC_IJDR3 data registers.
6	AWD_AH	W1C	0x0	Analog watchdog upper limit exceed interrupt flag: set by hardware when the sampled value exceeds AWD_HT; cleared by software writing 1.
5	AWD_UL	W1C	0x0	Analog watchdog lower limit drop interrupt flag: set by hardware when the sampled value drops below AWD_LT; cleared by software writing 1.
4:3	RSV	-	-	Reserved
2	OVR	W1C	0x0	Data overrun flag (OVR): set by hardware; cleared by

Bit	Name	Attribute	Reset Value	Description
				software writing 1. This flag is set by hardware when a new conversion result arrives before the previous conversion result in the DR register has been read. 0: No data overrun 1: Data overrun occurred
1	EOS	W1C	0x0	End of sequence (EOS): set when conversion of all enabled channels is completed; cleared by software writing 1.
0	EOC	W1C	0x0	End of single-shot conversion (EOC): set after the conversion of each channel is completed; cleared by software writing 1 or reading the ADC_DR0–ADC_DR7 and ADC_DR_OVS data registers.

21.4.2 ADC Interrupt Enable Register (ADC_IER) (Offset: 004H)

Bit	Name	Attribute	Reset Value	Description
31:9	RSV	-	-	Reserved
8	IJ_EOSIE	RW	0x0	Injected channel conversion sequence end interrupt enable register: 1 = Enabled, 0 = Disabled.
7	IJ_EOCIE	RW	0x0	Injected channel single-shot conversion end interrupt enable register: 1 = Enabled, 0 = Disabled.
6	AWD_AHIE	RW	0x0	Analog watchdog interrupt enable for sampled value exceeding upper limit: 1 = Enabled, 0 = Disabled.
5	AWD_ULIE	RW	0x0	Analog watchdog interrupt enable for sampled value below lower limit: 1 = Enabled, 0 = Disabled.
4:3	RSV	-	-	Reserved
2	OVRIE	RW	0x0	Data collision interrupt enable register: 1 = Enabled, 0 = Disabled.
1	EOSIE	RW	0x0	End of sequence interrupt enable register: 1 = Enabled, 0 = Disabled.
0	EOCIE	RW	0x0	End of conversion interrupt enable register: 1 = Enabled, 0 = Disabled.

21.4.3 ADC Control Register (ADC_CR) (Offset: 008H)

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	AWD_AH_EN	RW	0x0	Enable triggering of ATimer break when the analog watchdog sampled value exceeds the upper limit: 1: Enabled 0: Disabled
5	AWD_UL_EN	RW	0x0	Enable triggering of ATimer break when the analog watchdog sampled value drops below the lower limit: 1: Enabled 0: Disabled
4	BUFEN	RW	0x0	Enable analog input buffer: 0: Disabled 1: Enabled
3	STOP	W	0x0	ADC stop conversion register: Software writes 1 to stop continuous conversion mode.
2	IJSTART	RW	0x0	ADC injected channel start conversion register: Software writes 1 to start, automatically cleared by hardware.
1	START	RW	0x0	ADC start conversion register: Software writes 1 to start, automatically cleared by hardware.
0	ADEN	RW	0x0	ADC enable register:. Before starting conversion, ADEN must be set, and then wait for at least 5 MCLKs. 0: ADC disabled 1: ADC enabled

21.4.4 ADC Configuration Register (ADC_CFGR) (Offset: 00CH)

Bit	Name	Attribute	Reset Value	Description
31:30	RSV	-	-	Reserved
29:26	AWDCH	RW	0x0	Analog window watchdog monitoring channel selection (effective only when AWDSC = 1):

Bit	Name	Attribute	Reset Value	Description
				0000: AWD monitors SEQ0 0001: AWD monitors SEQ1 0010: AWD monitors SEQ2 0011: AWD monitors SEQ3 0100: AWD monitors SEQ4 0101: AWD monitors SEQ5 0110: AWD monitors SEQ6 0111: AWD monitors SEQ7 1000: AWD monitors IJSEQ0 1001: AWD monitors IJSEQ1 1010: AWD monitors IJSEQ2 1011: AWD monitors IJSEQ3 Others: Reserved
25	AWDSC	RW	0x0	Analog window watchdog sequence selection: 0: AWD monitors all enabled external input channels. 1: AWD monitors the single channel specified by AWDCH.
24	AWDEN	RW	0x0	Analog window watchdog enable register (AWD can only be enabled when START = 0): 0: AWD disabled 1: AWD enabled
23:20	OVSS	RW	0x0	Oversampling shift control register: 0000: No shift 0001: Shift right by 1 bit 0010: Shift right by 2 bits 0011: Shift right by 3 bits 0100: Shift right by 4 bits 0101: Shift right by 5 bits 0110: Shift right by 6 bits 0111: Shift right by 7 bits 1000: Shift right by 8 bits Others: Reserved
19:17	OVSR	RW	0x0	Oversampling rate control: 000: Sample 2 times 001: Sample 4 times 010: Sample 8 times

Bit	Name	Attribute	Reset Value	Description
				011: Sample 16 times 100: Sample 32 times 101: Sample 64 times 110: Sample 128 times 111: Sample 256 times
16	OVSEN	RW	0x0	Oversampling enable: 0: Oversampling disabled 1: Oversampling enabled
15	RSV	-	-	Reserved
14	IOTRFEN	RW	0x0	Pin trigger signal digital filter enable: 0: Digital filter disabled 1: Digital filter enabled
13:12	TRGCFG	RW	0x0	Hardware trigger signal enable and polarity selection: 00: Hardware trigger disabled 01: Rising edge trigger 10: Falling edge trigger 11: Both rising and falling edge trigger
11	SEMI	RW	0x0	Single-shot conversion semi-automatic mode (effective only when CONT = 0): 0: Automatic mode 1: Semi-automatic mode
10	WAIT	RW	0x0	Wait mode control: 0: No wait, overrun may occur if the previous conversion data is not read in time. 1: Wait mode, the next conversion will not start until the previous conversion data is read.
9	CONT	RW	0x0	Continuous conversion mode enable: 0: Single-shot conversion 1: Continuous conversion
8	OVRM	RW	0x0	Overrun mode control: 0: Keep the previous data and discard the current conversion value when overrun occurs. 1: Overwrite the previous data when overrun occurs.
7:1	RSV	-	-	Reserved
0	DMAEN	RW	0x0	DMA enable:

Bit	Name	Attribute	Reset Value	Description
				0: DMA disabled 1: DMA enabled

21.4.5 ADC Sampling Time Control Register (ADC_SMTR) (Offset: 010H)

Bit	Name	Attribute	Reset Value	Description
31:21	RSV	-	-	Reserved
20:16	CKD	RW	0x0	ADCCLK prescaler register: 0x00, 0x01: System clock frequency divided by 2 0x02: System clock frequency divided by 3 …… 0x1F: System clock frequency divided by 32
15:12	RSV	-	-	Reserved
11:8	CHCG	RW	0x8	ADC sampling channel switching wait time: 0000, 0001, 0010: 2 * t_{ADCCLK} 0011: 3 * t_{ADCCLK} 0100: 4 * t_{ADCCLK} 0101: 5 * t_{ADCCLK} 0110: 6 * t_{ADCCLK} 0111: 7 * t_{ADCCLK} 1000: 8 * t_{ADCCLK} 1001: 9 * t_{ADCCLK} 1010: 10 * t_{ADCCLK} Others: 11 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.
7:0	RSV	-	-	Reserved

21.4.6 ADC Channel Control Register (ADC_CHER) (Offset: 014H)

Bit	Name	Attribute	Reset Value	Description
31:22	RSV	-	-	Reserved
21	CH_EN	R	0x1	Analog channel enable: 0: Enabled

Bit	Name	Attribute	Reset Value	Description
				1: Disabled
20:17	CUR	RW	0x0	BUFF and CMP bias current configuration
16	RDTEST	RW	0x1	Write '0' to enter RD test mode.
15:11	CAPTEST	RW	0x0	1.5 V input signal capacitance mismatch selection (for RD test)
10:9	VCMSEL	RW	0x0	CMP VCM reference voltage selection
8:7	AZDELY	RW	0x0	1.5 V input signal delay selection
6:5	CMPOPT	RW	0x0	CMP gain selection
4:3	VREFOPT	RW	0x0	ADC positive reference voltage selection: 0: Disabled (floating) 1: VREFPAVDD 2: VREFPLDO 3: PD3
2:0	SEQNUM	RW	0x0	Total length setting bit of ADC channel sequence: The total number of analog channels in one AD conversion process is SEQNUM + 1.

21.4.7 ADC Channel Selection Register (ADC_SEQ) (Offset: 018H)

Bit	Name	Attribute	Reset Value	Description
31:28	SEQ7	RW	0x0	Analog channel 7 selection register group, specifying the corresponding analog input channel respectively.
27:24	SEQ6	RW	0x0	Analog channel 6 selection register group, specifying the corresponding analog input channel respectively.
23:20	SEQ5	RW	0x0	Analog channel 5 selection register group, specifying the corresponding analog input channel respectively.
19:16	SEQ4	RW	0x0	Analog channel 4 selection register group, specifying the corresponding analog input channel respectively.
15:12	SEQ3	RW	0x0	Analog channel 3 selection register group, specifying the corresponding analog input channel respectively.

Bit	Name	Attribute	Reset Value	Description
11:8	SEQ2	RW	0x0	Analog channel 2 selection register group, specifying the corresponding analog input channel respectively.
7:4	SEQ1	RW	0x0	Analog channel 1 selection register group, specifying the corresponding analog input channel respectively.
3:0	SEQ0	RW	0x0	Analog channel 0 selection register group, specifying the corresponding analog input channel respectively.

21.4.8 ADC Analog Watchdog Threshold Register (ADC_HLTR) (Offset: 020H)

Bit	Name	Attribute	Reset Value	Description
31:16	AWD_HT	RW	0x0	AWD monitoring high threshold
15:0	AWD_LT	RW	0x0	AWD monitoring low threshold

21.4.9 ADC Injected Channel Control Register (ADC_IJCHER) (Offset: 024H)

Bit	Name	Attribute	Reset Value	Description
31:15	RSV	-	-	Reserved
14	IJIOTRFEN	RW	0x0	Digital filter enable for injected channel pin trigger signal: 0: Digital filter disabled 1: Digital filter enabled
13:12	IJTRGCFG	RW	0x0	Injected channel trigger signal enable and polarity selection: 00: Trigger disabled 01: Rising edge trigger 10: Falling edge trigger 11: Both rising and falling edge trigger
11:5	RSV	-	-	Reserved
4	IJ_SEMI	RW	0x0	ADC injected channel conversion mode

Bit	Name	Attribute	Reset Value	Description
				selection bit: 0: One trigger converts all channels. 1: One trigger converts one channel. Note: This function is only valid when ADC_CFGR.CONT = 0 and ADC_CFGR.SEMI = 1.
3	IJ_AUTO	RW	0x0	ADC injected channel automatic mode selection bit: 0: Automatic mode disabled 1: Automatic mode enabled
2:1	IJ_SEQNUM	RW	0x0	Total length setting bit of ADC injected channel sequence: The total number of analog channels in one AD conversion process is IJ_SEQNUM + 1.
0	IJ_EN	RW	0x0	ADC injected channel enable: 0: Injected channel disabled 1: Injected channel enabled

21.4.10 ADC Injected Channel Selection Register (ADC_IJSEQ) (Offset: 028H)

Bit	Name	Attribute	Reset Value	Description
31:16	RSV	-	-	Reserved
15:12	IJ_SEQ3	RW	0x0	Analog channel 3 selection register group, specifying the corresponding analog input channel respectively.
11:8	IJ_SEQ2	RW	0x0	Analog channel 2 selection register group, specifying the corresponding analog input channel respectively.
7:4	IJ_SEQ1	RW	0x0	Analog channel 1 selection register group, specifying the corresponding analog input channel respectively.
3:0	IJ_SEQ0	RW	0x0	Analog channel 0 selection register group, specifying the corresponding analog input channel respectively.

21.4.11 ADC Data Register 0 (ADC_DR0) (Offset: 030H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC result register 0

21.4.12 ADC Data Register 1 (ADC_DR1) (Offset: 034H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC result register 1

21.4.13 ADC Data Register 2 (ADC_DR2) (Offset: 038H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC result register 2

21.4.14 ADC Data Register 3 (ADC_DR3) (Offset: 03CH)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC result register 3

21.4.15 ADC Data Register 4 (ADC_DR4) (Offset: 040H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC result register 4

21.4.16 ADC Data Register 5 (ADC_DR5) (Offset: 044H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC result register 5

21.4.17 ADC Data Register 6 (ADC_DR6) (Offset: 048H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC result register 6

21.4.18 ADC Data Register 7 (ADC_DR7) (Offset: 04CH)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC result register 7

21.4.19 ADC Injected Channel Data Register 0 (ADC_IJDR0) (Offset: 050H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC injected channel result register 0

21.4.20 ADC Injected Channel Data Register 1 (ADC_IJDR1) (Offset: 054H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC injected channel result register 1

21.4.21 ADC Injected Channel Data Register 2 (ADC_IJDR2) (Offset: 058H)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC injected channel result register 2

21.4.22 ADC Injected Channel Data Register 3 (ADC_IJDR3) (Offset: 05CH)

Bit	Name	Attribute	Reset Value	Description
31:12	RSV	-	-	Reserved
11:0	DATA	R	0x0	ADC injected channel result register 3

21.4.23 ADC Oversampling Average Result Register (ADC_DR_OVS) (Offset: 060H)

Bit	Name	Attribute	Reset Value	Description
31:20	RSV	-	-	Reserved
19:0	DATA	R	0x0	ADC oversampling average result

21.4.24 ADC Hardware Trigger Enable Register (ADC_HDT) (Offset: 0A8H)

Bit	Name	Attribute	Reset Value	Description
31:21	RSV	-	-	Reserved
20	ATIMER_OC45TRGO_HT	RW	0x0	ATIMER OC4/OC5 TRGO hardware trigger enable: 1: Enabled 0: Disabled
19	ATIMER_OC5TRGO_HT	RW	0x0	ATIMER OC5 TRGO hardware trigger enable: 1: Enabled 0: Disabled
18	ATIMER_OC4TRGO_HT	RW	0x0	ATIMER OC4 TRGO hardware trigger enable: 1: Enabled 0: Disabled

Bit	Name	Attribute	Reset Value	Description
17	PD3_HT	RW	0x0	PD3 hardware trigger enable: 1: Enabled 0: Disabled
16	PD1_HT	RW	0x0	PD1 hardware trigger enable: 1: Enabled 0: Disabled
15	PD0_HT	RW	0x0	PD0 hardware trigger enable: 1: Enabled 0: Disabled
14	PC4_HT	RW	0x0	PC4 hardware trigger enable: 1: Enabled 0: Disabled
13	PC3_HT	RW	0x0	PC3 hardware trigger enable: 1: Enabled 0: Disabled
12	PC2_HT	RW	0x0	PC2 hardware trigger enable: 1: Enabled 0: Disabled
11	RSV	-	-	Reserved
10	LVD_HT	RW	0x0	LVD hardware trigger enable: 1: Enabled 0: Disabled
9	LPTIMER_HT2	RW	0x0	LPTIMER OUT2 hardware trigger enable: 1: Enabled 0: Disabled
8	LPTIMER_HT1	RW	0x0	LPTIMER OUT1 hardware trigger enable: 1: Enabled 0: Disabled
7	GTIMER2_TRGO_HT	RW	0x0	GTIMER2 TRGO hardware trigger enable: 1: Enabled 0: Disabled

Bit	Name	Attribute	Reset Value	Description
6	GTIMER1_TRGO_HT	RW	0x0	GTIMER1 TRGO hardware trigger enable: 1: Enabled 0: Disabled
5	GTIMER0_TRGO_HT	RW	0x0	GTIMER0 TRGO hardware trigger enable: 1: Enabled 0: Disabled
4	CMP1_HT	RW	0x0	CMP1 hardware trigger enable: 1: Enabled 0: Disabled
3	CMP0_HT	RW	0x0	CMP0 hardware trigger enable: 1: Enabled 0: Disabled
2	ATIMER_OC5REF_HT	RW	0x0	ATIMER OC5 REF hardware trigger enable: 1: Enabled 0: Disabled
1	ATIMER_OC4REF_HT	RW	0x0	ATIMER OC4 REF hardware trigger enable: 1: Enabled 0: Disabled
0	ATIMER_TRGO_HT	RW	0x0	ATIMER TRGO hardware trigger enable: 1: Enabled 0: Disabled

21.4.25 ADC Injected Channel Hardware Trigger Enable Register (ADC_IJHDT) (Offset: 0ACH)

Bit	Name	Attribute	Reset Value	Description
31:21	RSV	-	-	Reserved
20	ATIMER_OC45TRGO_IJHT	RW	0x0	ATIMER OC4/OC5 TRGO hardware

Bit	Name	Attribute	Reset Value	Description
				trigger enable: 1: Enabled 0: Disabled
19	ATIMER_OC5TRGO_IJHT	RW	0x0	ATIMER OC5 TRGO hardware trigger enable: 1: Enabled 0: Disabled
18	ATIMER_OC4TRGO_IJHT	RW	0x0	ATIMER OC4 TRGO hardware trigger enable: 1: Enabled 0: Disabled
17	PD3_IJHT	RW	0x0	PD3 hardware trigger enable: 1: Enabled 0: Disabled
16	PD1_IJHT	RW	0x0	PD1 hardware trigger enable: 1: Enabled 0: Disabled
15	PD0_IJHT	RW	0x0	PD0 hardware trigger enable: 1: Enabled 0: Disabled
14	PC4_IJHT	RW	0x0	PC4 hardware trigger enable: 1: Enabled 0: Disabled
13	PC3_IJHT	RW	0x0	PC3 hardware trigger enable: 1: Enabled 0: Disabled
12	PC2_IJHT	RW	0x0	PC2 hardware trigger enable: 1: Enabled 0: Disabled
11	RSV	-	-	Reserved
10	LVD_IJHT	RW	0x0	LVD hardware trigger enable: 1: Enabled 0: Disabled
9	LPTIMER_IJHT2	RW	0x0	LPTIMER OUT2 hardware trigger enable: 1: Enabled

Bit	Name	Attribute	Reset Value	Description
				0: Disabled
8	LPTIMER_IJHT1	RW	0x0	LPTIMER OUT1 hardware trigger enable: 1: Enabled 0: Disabled
7	GTIMER2_TRGO_IJHT	RW	0x0	GTIMER2 TRGO hardware trigger enable: 1: Enabled 0: Disabled
6	GTIMER1_TRGO_IJHT	RW	0x0	GTIMER1 TRGO hardware trigger enable: 1: Enabled 0: Disabled
5	GTIMER0_TRGO_IJHT	RW	0x0	GTIMER0 TRGO hardware trigger enable: 1: Enabled 0: Disabled
4	CMP1_IJHT	RW	0x0	CMP1 hardware trigger enable: 1: Enabled 0: Disabled
3	CMP0_IJHT	RW	0x0	CMP0 hardware trigger enable: 1: Enabled 0: Disabled
2	ATIMER_OC5REF_IJHT	RW	0x0	ATIMER OC5 REF hardware trigger enable: 1: Enabled 0: Disabled
1	ATIMER_OC4REF_IJHT	RW	0x0	ATIMER OC4 REF hardware trigger enable: 1: Enabled 0: Disabled
0	ATIMER_TRGO_IJHT	RW	0x0	ATIMER TRGO hardware trigger enable: 1: enabled 0: Disabled

21.4.26 ADC Channel Sampling Time Configuration Register 0 (ADC_SMTR0) (Offset: 0B0H)

Bit	Name	Attribute	Reset Value	Description
31:24	SMTR3	RW	0x0	Used to configure the sampling time of analog channel 3: (ADC_SMTR3 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.
23:16	SMTR2	RW	0x0	Used to configure the sampling time of analog channel 2: (ADC_SMTR2 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.
15:8	SMTR1	RW	0x0	Used to configure the sampling time of analog channel 1: (ADC_SMTR1 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.
7:0	SMTR0	RW	0x0	Used to configure the sampling time of analog channel

Bit	Name	Attribute	Reset Value	Description
				0: (ADC_SMTR0 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.

21.4.27 ADC Channel Sampling Time Configuration Register 1 (ADC_SMTR1) (Offset: 0B4H)

Bit	Name	Attribute	Reset Value	Description
31:24	SMTR7	RW	0x0	Used to configure the sampling time of analog channel 7: (ADC_SMTR7 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.
23:16	SMTR6	RW	0x0	Used to configure the sampling time of analog channel 6: (ADC_SMTR6 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.

Bit	Name	Attribute	Reset Value	Description
15:8	SMTR5	RW	0x0	Used to configure the sampling time of analog channel 5: (ADC_SMTR5 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.
7:0	SMTR4	RW	0x0	Used to configure the sampling time of analog channel 4: (ADC_SMTR4 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.

21.4.28 ADC Injected Channel Sampling Time Configuration Register (ADC_IJSMTR) (Offset: 0B8H)

Bit	Name	Attribute	Reset Value	Description
31:24	IJSMTR3	RW	0x0	Used to configure the sampling time of injected channel 3: (ADC_IJSMTR3 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK}

Bit	Name	Attribute	Reset Value	Description
				Note: t_{ADCCLK} is the ADCCLK clock cycle.
23:16	IJSMTR2	RW	0x0	Used to configure the sampling time of injected channel 2: (ADC_IJSMTR2 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.
15:8	IJSMTR1	RW	0x0	Used to configure the sampling time of injected channel 1: (ADC_IJSMTR1 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.
7:0	IJSMTR0	RW	0x0	Used to configure the sampling time of injected channel 0: (ADC_IJSMTR0 + 2) * t_{ADCCLK}, i.e.: 8'd0: 2 * t_{ADCCLK} 8'd1: 3 * t_{ADCCLK} 8'd2: 4 * t_{ADCCLK} … 8'd253: 255 * t_{ADCCLK} 8'd254: 256 * t_{ADCCLK} 8'd255: 257 * t_{ADCCLK} Note: t_{ADCCLK} is the ADCCLK clock cycle.

21.5 ADC Operation Procedure

21.5.1 Single-channel A/D Conversion in Single-shot Scan Mode

In single-shot scan mode, ADC performs only one conversion after starting the conversion.

1. Configure ADC_CR.ADEN to 0 to release the analog ADC reset and power it on.
2. Configure ADC_SMTR.CDK to set the ADC clock division ratio.
3. Configure ADC_SMTR.CHCG and ADC_SMTR0/1 to set the ADC conversion speed.
4. Configure ADC_CHER.VREFOPT to set the ADC reference voltage source.
5. Configure ADC_CFGR.CONT to 0 to select the single-shot scan mode.
6. Configure ADC_CFGR.SEMI to select the single trigger mode (semi-automatic/full-automatic).
7. Configure ADC_CHER.SEQNUM to 0 to set the scan sequence to a single channel.
8. Configure ADC_SEQ.SEQ0 to set the analog input channel corresponding to the channel.
9. Configure the channel to be converted as an analog interface (PAD_ADS) based on the GPIO pin corresponding to the ADC input channel.
10. Configure ADC_CR.ADEN to 1 to enable the ADC module.
11. Configure ADC_CR.START to 1 to start the ADC conversion.
12. Wait for the ADC_ISR.EOC single-shot conversion end flag to be 1, and read the data in the ADC_DR of the corresponding channel.
13. If multiple single-shot conversions are required, repeat steps 11 and 12.
14. If the ADC_IER.EOCIE and ADC_IER.EOSIE interrupts are enabled, wait for the interrupt to be triggered and then read ADC_DR.

21.5.2 Multi-channel A/D Conversion in Single-shot Scan Mode

In single-shot scan mode, when multiple channels are enabled simultaneously, after starting the ADC conversion, channels SEQ0–SEQ7 will be sequentially converted. The input channels corresponding to SEQ0–SEQ7 can be configured. When the conversion of channel SEQ7 is completed, it indicates that the single-shot scan conversion is finished.

1. Configure ADC_CR.ADEN to 0 to release the analog ADC reset and power it on.
2. Configure ADC_SMTR.CDK to set the ADC clock division ratio.
3. Configure ADC_SMTR.CHCG and ADC_SMTR0/1 to set the ADC conversion speed.
4. Configure ADC_CHER.VREFOPT to set the ADC reference voltage source.
5. Configure ADC_CFGR.CONT to 0 to select the single-shot scan mode.
6. Configure ADC_CFGR.SEMI to select the single-shot conversion trigger mode (semi-automatic/full-automatic).
7. Configure ADC_CHER.SEQNUM to n to set the number of channels in the scan sequence.
8. Configure ADC_SEQ to set the analog input channels corresponding to each channel (i.e., set channel priority).
9. Configure the channel to be converted as an analog interface (PAD_ADS) based on the GPIO pin corresponding to the ADC input channel.
10. Configure ADC_CR.ADEN to 1 to enable the ADC module.
11. Configure ADC_CR.START to 1 to start the ADC conversion.
12. Wait for the ADC_ISR.EOC single-shot conversion end flag to be 1, and read the data in the ADC_DR of the corresponding channel.
13. Repeat the above operations (if single-shot conversion semi-automatic trigger mode is

configured, repeat steps 11 and 12; for full-automatic trigger, only repeat step 12) until the ADC_ISR.EOS conversion sequence end flag bit is set to 1, indicating that all channels in the single-shot conversion have been converted.

14. If multiple single-shot conversions are required, repeat steps 12, 13 and 14.
15. If the ADC_IER.EOCIE and ADC_IER.EOSIE interrupts are enabled, wait for the interrupt to be triggered and then read ADC_DR.

Notes:

- In multi-channel conversion, it is recommended to wait until the ADC_ISR.EOS conversion sequence end flag is valid before reading the data from ADC_DR of each channel.
- In multi-channel conversion, if the sampling accuracy is insufficient, it is recommended to appropriately modify the values of ADC_SMTR.CHCG and ADC_SMTR0/1.

21.5.3 Single-channel A/D Conversion in Continuous Scan Mode

In continuous scan mode, after starting a single ADC conversion, the selected channel will be continuously converted. Writing 1 to ADC_CR.STOP can stop the conversion.

1. Configure ADC_CR.ADEN to 0 to release the analog ADC reset and power it on.
2. Configure ADC_SMTR.CDK to set the ADC clock division ratio.
3. Configure ADC_SMTR.CHCG and ADC_SMTR0/1 to set the ADC conversion speed.
4. Configure ADC_CHER.VREFOPT to set the ADC reference voltage source.
5. Configure ADC_CFGR.CONT to 1 to select the continuous scan mode.
6. Configure ADC_CHER.SEQNUM to 0 to set the scan sequence to a single channel.
7. Configure ADC_SEQ.SEQ0 to set the analog input channel corresponding to the channel.
8. Configure the channel to be converted as an analog interface (PAD_ADS) based on the GPIO

pin corresponding to the ADC input channel.

9. Configure ADC_CR.ADEN to 1 to enable the ADC module.
10. Configure ADC_CR.START to 1 to start the ADC conversion.
11. Wait for the ADC_ISR.EOC single-shot conversion end flag to be 1, and read the data in the ADC_DR of the corresponding channel.
12. To read multiple ADC data, repeat the previous step.
13. If the ADC_IER.EOCIE and ADC_IER.EOSIE interrupts are enabled, wait for the interrupt to be triggered and then read ADC_DR.

21.5.4 Multi-channel A/D Conversion in Continuous Scan Mode

In continuous scan mode, after starting a single ADC conversion, the selected channel will be continuously converted. Writing 1 to ADC_CR.STOP can stop the conversion. When multiple channels are enabled simultaneously, conversions will be performed sequentially from channels SEQ0 to SEQ7. The input channels corresponding to SEQ0–SEQ7 can be configured. When the conversion of channel SEQ7 is completed, it indicates that the single-shot scan conversion is finished.

1. Configure ADC_CR.ADEN to 0 to release the analog ADC reset and power it on.
2. Configure ADC_SMTR.CDK to set the ADC clock division ratio.
3. Configure ADC_SMTR.CHCG and ADC_SMTR0/1 to set the ADC conversion speed.
4. Configure ADC_CHER.VREFOPT to set the ADC reference voltage source.
5. Configure ADC_CFGR.CONT to 1 to select the continuous scan mode.
6. Configure ADC_CHER.SEQNUM to n to set the number of channels in the scan sequence.
7. Configure ADC_SEQ to set the analog input channels corresponding to each channel (i.e.,

set channel priority).

8. Configure the channel to be converted as an analog interface (PAD_ADS) based on the GPIO pin corresponding to the ADC input channel.
9. Configure ADC_CR.ADEN to 1 to enable the ADC module.
10. Configure ADC_CR.START to 1 to start the ADC conversion.
11. Wait for the ADC_ISR.EOC single-shot conversion end flag to be 1, and read the data in the ADC_DR of the corresponding channel.
12. To read multiple ADC data, repeat the above step.
13. If the ADC_IER.EOCIE and ADC_IER.EOSIE interrupts are enabled, wait for the interrupt to be triggered and then read ADC_DR.

Notes:

- In multi-channel conversion, it is recommended to wait until the ADC_ISR.EOS conversion sequence end flag is valid before reading the data from ADC_DR of each channel.
- In multi-channel conversion, if the sampling accuracy is insufficient, it is recommended to appropriately modify the values of ADC_SMTR.CHCG and ADC_SMTR0/1.

21.5.5 A/D Conversion Initiated by Hardware-triggered Event

A hardware-triggered event can only trigger the single-shot scan mode. After the ADC starts the conversion, it performs the conversion only once.

1. Configure ADC_CR.ADEN to 0 to release the analog ADC reset and power it on.
2. Configure ADC_SMTR.CDK to set the ADC clock division ratio.
3. Configure ADC_SMTR.CHCG and ADC_SMTR0/1 to set the ADC conversion speed.
4. Configure ADC_CHER.VREFOPT to set the ADC reference voltage source.

5. Configure ADC_CFGR.CONT to 0 to select the single-shot scan mode.
6. Configure ADC_CFGR.SEMI to select the single trigger mode (semi-automatic/full-automatic).
7. Configure ADC_CHER.SEQNUM to 0 to set the scan sequence to a single channel.
8. Configure ADC_SEQ.SEQ0 to set the analog input channel corresponding to the channel.
9. Configure the channel to be converted as an analog interface (PAD_ADS) based on the GPIO pin corresponding to the ADC input channel.
10. Configure ADC_CR.ADEN to 1 to enable the ADC module.
11. Initialize the hardware that needs to act as the trigger source for the ADC conversion.
12. Configure the corresponding hardware trigger event in ADC_HDT to 1 to enable hardware triggering.
13. Configure ADC_CFGR.IOTRFEN to enable digital filtering for the trigger signal.
14. Configure ADC_CFGR.TRGCFG to select the polarity of the hardware trigger event.
15. Wait for the ADC_ISR.EOC single-shot conversion end flag to be 1, and read the data in the ADC_DR of the corresponding channel.
16. If multiple single-shot conversions are required, trigger the hardware event multiple times.
17. If the ADC_IER.EOCIE and ADC_IER.EOSIE interrupts are enabled, wait for the interrupt to be triggered and then read ADC_DR.

21.5.6 A/D Conversion in Normal Injection Mode of Injected Channels

The ADC can inject up to four conversion channels. The conversion of injected channels is similar

to normal conversion, and the results are stored in ADC_IJDR0–ADC_IJDR3.

1. Configure ADC_CR.ADEN to 0 to release the analog ADC reset and power it on.
2. Configure ADC_SMTR.CDK to set the ADC clock division ratio.
3. Configure ADC_SMTR.CHCG and ADC_IJSMTR to set the ADC conversion speed.
4. Configure ADC_CHER.VREFOPT to set the ADC reference voltage source.
5. Configure ADC_CFGR.CONT to 0 to select the single-shot scan mode.
6. Configure ADC_CFGR.SEMI to select the single-shot conversion trigger mode (semi-automatic/full-automatic).
7. Configure ADC_CHER.SEQNUM to n to set the number of channels in the scan sequence.
8. Configure ADC_SEQ to set the analog input channels corresponding to each channel (i.e., set channel priority).
9. Configure ADC_IJCHER.IJ_AUTO to 0 to set the injected channels to normal injection mode.
10. Configure ADC_IJCHER.IJ_SEQNUM to n to set the number of channels in the injected channel scan sequence.
11. Configure ADC_IJSEQ to set the analog input channels corresponding to each injected channel (i.e., set channel priority).
12. Configure ADC_IJCHER.IJ_EN to 1 to enable the injected channels.
13. Configure the channel to be converted as an analog interface (PAD_ADS) based on the GPIO pin corresponding to the ADC input channel.
14. Configure ADC_CR.ADEN to 1 to enable the ADC module.
15. Configure ADC_CR.START to 1 to start the ADC conversion.
16. Wait for the ADC_ISR.EOC single-shot conversion end flag to be 1, and read the data in the

ADC_DR of the corresponding channel.

17. Configure ADC_CR.IJSTART to 1 to start the ADC injected channel conversion.
18. Wait for the ADC_ISR.IJ_EOC single-shot conversion end flag to be 1, and read the data in the ADC_IJDR of the corresponding channel.
19. Repeat the above operations until the ADC_ISR.EOS and ADC_ISR.IJ_EOS conversion sequence end flag bits are set to 1, indicating that all regular channels and injected channels in the single-shot conversion have been converted.
20. If the ADC_IER.EOCIE and ADC_IER.EOSIE interrupts are enabled, wait for the interrupt to be triggered and then read ADC_DR. If the ADC_IER.IJ_EOCIE and ADC_IER.IJ_EOSIE interrupts are enabled, wait for the interrupt to be triggered and then read the ADC_IJDR.

21.5.7 A/D Conversion in Automatic Injection Mode of Injected Channels

The ADC can inject up to four conversion channels. The conversion of injected channels is similar to normal conversion, and the results are stored in ADC_IJDR0–ADC_IJDR3.

1. Configure ADC_CR.ADEN to 0 to release the analog ADC reset and power it on.
2. Configure ADC_SMTR.CDK to set the ADC clock division ratio.
3. Configure ADC_SMTR.CHCG and ADC_IJSMTR to set the ADC conversion speed.
4. Configure ADC_CHER.VREFOPT to set the ADC reference voltage source.
5. Configure ADC_CFGR.CONT to 0 to select the single-shot scan mode.
6. Configure ADC_CFGR.SEMI to select the single-shot conversion trigger mode (semi-automatic/full-automatic).
7. Configure ADC_CHER.SEQNUM to n to set the number of channels in the scan sequence.

8. Configure ADC_SEQ to set the analog input channels corresponding to each channel (i.e., set channel priority).
9. Configure ADC_IJCHER.IJ_AUTO to 1 to set the injected channels to automatic injection mode.
10. Configure ADC_IJCHER.IJ_SEQNUM to n to set the number of channels in the injected channel scan sequence.
11. Configure ADC_IJSEQ to set the analog input channels corresponding to each injected channel (i.e., set channel priority).
12. Configure ADC_IJCHER.IJ_EN to 1 to enable the injected channels.
13. Configure the channel to be converted as an analog interface (PAD_ADS) based on the GPIO pin corresponding to the ADC input channel.
14. Configure ADC_CR.ADEN to 1 to enable the ADC module.
15. Configure ADC_CR.START to 1 to start the ADC conversion.
16. Wait for the ADC_ISR.EOC single-shot conversion end flag to be 1, and read the data in the ADC_DR of the corresponding channel.
17. Configure ADC_CR.IJSTART to 1 to start the ADC injected channel conversion.
18. Repeat the above operations until the ADC_ISR.EOS and ADC_ISR.IJ_EOS conversion sequence end flag bits are set to 1, indicating that all channels in the single-shot conversion have been converted.
19. Wait for the ADC_ISR.IJ_EOC single-shot conversion end flag to be 1, and read the data in the ADC_IJDR of the corresponding channel.
20. Repeat the above operations until the ADC_ISR.IJ_EOS conversion sequence end flag bit is set to 1, indicating that all injected channels in the single-shot conversion have been

converted.

21. If the ADC_IER.EOCIE and ADC_IER.EOSIE interrupts are enabled, wait for the interrupt to be triggered and then read ADC_DR. If the ADC_IER.IJ_EOCIE and ADC_IER.IJ_EOSIE interrupts are enabled, wait for the interrupt to be triggered and then read the ADC_IJDR.

21.5.8 Notes

- After each channel is sampled, the data is stored in the ADC_DR0–ADC_DR7 registers. The software must read the data in time before the next conversion, or transfer the data via DMA. Failure to read the data in time will cause an overrun, setting the overrun flag and potentially generating an interrupt. The ADC controller supports automatic waiting: if the WAIT register is set by software, the ADC controller will not initiate a new conversion until the ADC_DATA register is read. Hardware trigger events arriving during the waiting state will also be ignored. The WAIT register is also effective in DMA mode, meaning that if DMA has not read the previous conversion result, the ADC controller will not start a new conversion.
- The overrun mode and wait mode also apply to the data of injected channels (ADC_IJDR0–ADC_IJDR3).
- DMA only supports data transfer for regular channels (ADC_DR0–ADC_DR7), and does not support data transfer for injected channels.
- Each hardware trigger event triggers only one ADC conversion (i.e., single sampling mode). When enabling hardware triggering, it is recommended to configure the ADC mode as single-shot scan mode.

21.6 Workflow of ADC Sampling via PGA Buffer

21.6.1 Diagram of ADC Sampling via PGA Buffer

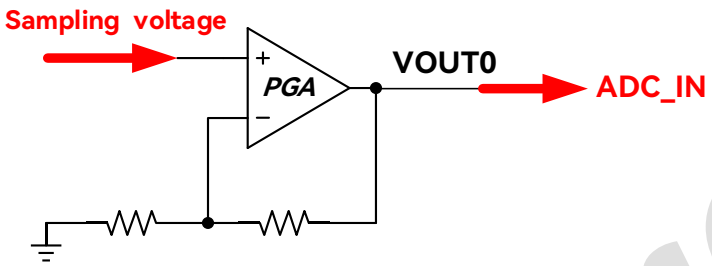


Figure 21-1: Diagram of ADC Sampling via PGA Buffer

21.6.2 Flowchart of ADC Sampling via PGA Buffer

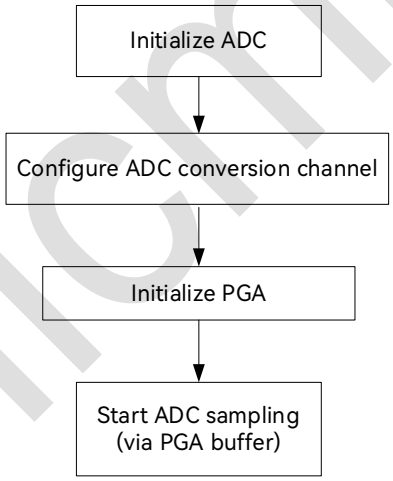


Figure 21-2: Flowchart of ADC Sampling via PGA Buffer

21.6.3 Workflow of ADC Sampling via PGA Buffer

1. Initialize the ADC. Select ADC reference voltage source, ADCCLK division factor and sampling mode.

2. Configure ADC conversion channels.

3. Initialize the PGA. Configure the PGA such that its positive terminal selects the sampling voltage (external input or AVSS), and its negative terminal is connected to the output internally, enabling the ADC input channel to be buffered via PGA.

4. Start ADC sampling.

22 COMP

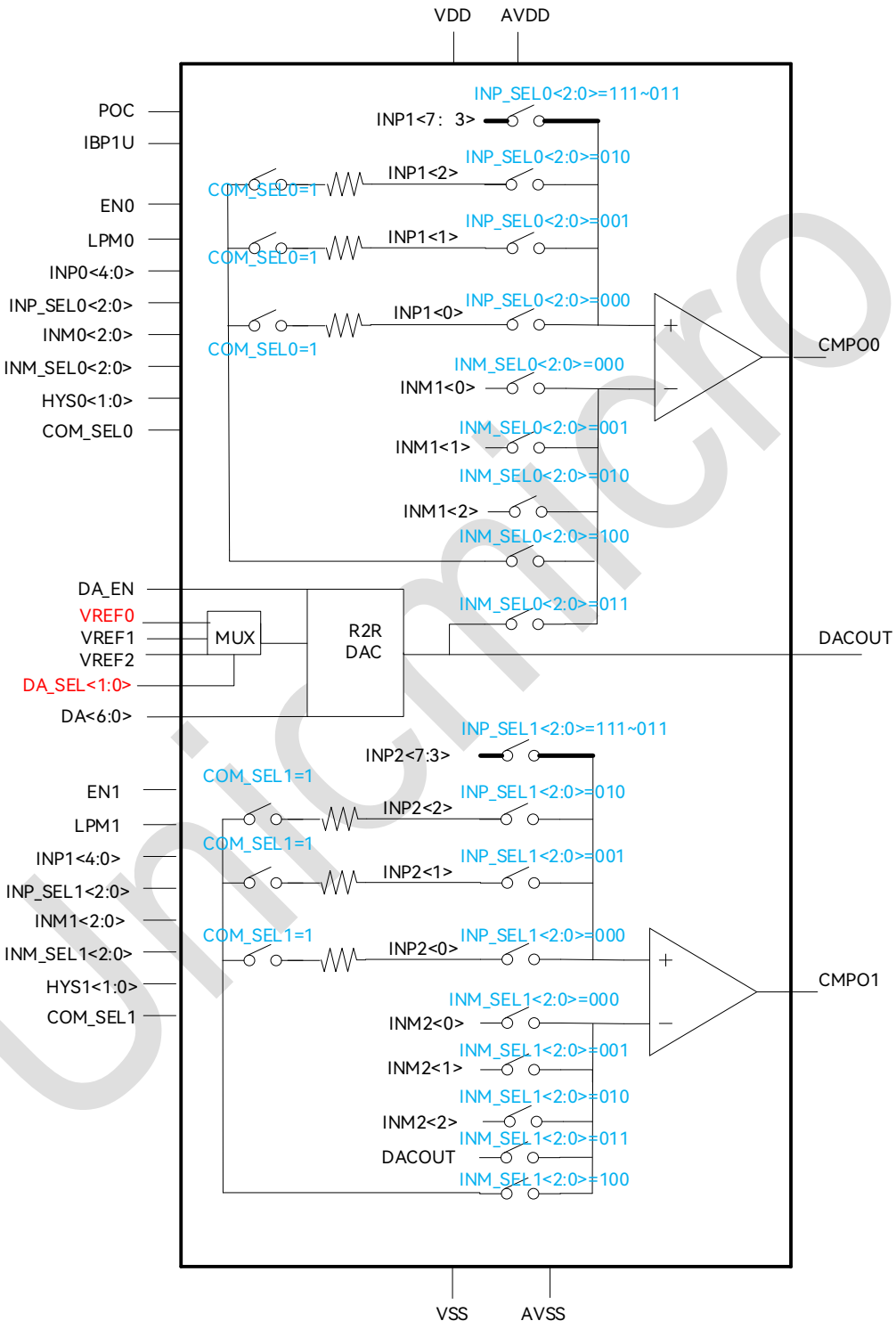
22.1 Overview

COMP is a hysteresis comparator with rail-to-rail input, and its input terminal can be configured as required. COMP can be used as a voltage comparator with two input terminals (IN+ and IN-), allowing one of the input terminals to be selected as a reference point for comparison. When the voltage at the other input terminal is lower than the reference voltage, the comparator outputs a low level; otherwise, it outputs a high level.

22.2 Main Features

- 2 x voltage comparators
- Capable of generating comparison interrupts
- Adjustable comparator accuracy
- Comparison results can be output via Pad

22.3 COMP Functional Block Diagram



Note: The texts in blue indicate the conditions for the switch to close.

Figure 22-1: COMP Analog Functional Block Diagram

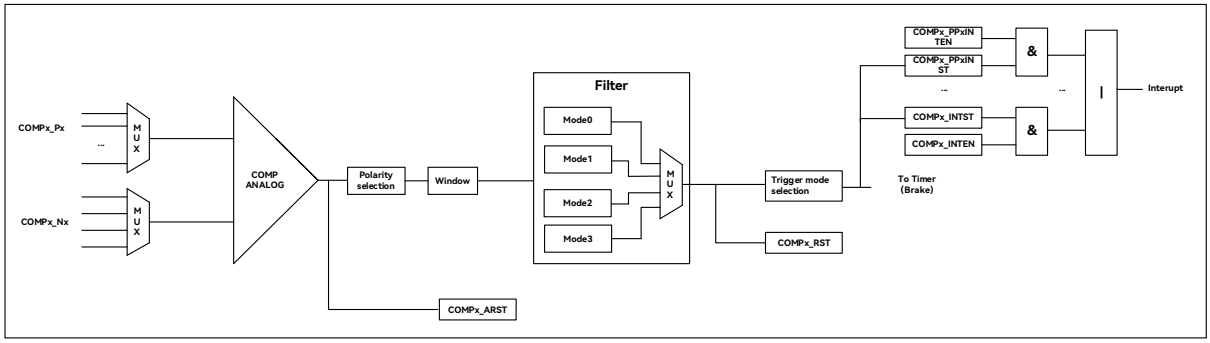


Figure 22-2: COMP Overall Functional Block Diagram

22.4 COMP Filtering Modes

In applications, only one filtering mode can be selected at a time; two or more filtering modes cannot be used in parallel. The filtering time for all four filter modes is set by the COMP0_LVTSET.COMP0_LVTSET register. **(The CNT bit is set and cleared by software.)**

1. Mode 1

When the output of the analog comparator (after polarity selection) is 1, the counter starts counting. When the counter value equals the value set in the COMP0_LVTSET register, the filter result is 1 (i.e., the current comparator result is valid). When the output of the analog comparator (after polarity selection) is 0, the counter value is cleared to 0.

2. Mode 2

When the output of the analog comparator (after polarity selection) is 1, the counter starts counting. When the counter value equals the value set in the COMP0_LVTSET register, the filter result is 1 (i.e., the current comparator result is valid). When the output of the analog comparator (after polarity selection) is 0, the counter value remains unchanged.

3. Mode 3

When the output of the analog comparator (after polarity selection) is 1, the counter starts counting. When the counter value equals the value set in the COMP0_LVTSET register, the filter

result is 1 (i.e., the current comparator result is valid). When the output of the analog comparator (after polarity selection) is 0, the counter value decreases by 1 each time until it reaches 0.

4. Mode 4

When the output of the analog comparator (after polarity selection) is 1, the counter starts counting. When the counter value equals the value set in the COMP0_LVTSET register, the filter result is 1 (i.e., the current comparator result is valid). When the output of the analog comparator (after polarity selection) is 0, the counter value decreases by 2 each time until it reaches 0.

22.5 Window Function

By configuring the WINCFG register to select an edge of the PWM signal, the comparator result will be invalid for a certain period after this edge and will not participate in the subsequent filtering or result calculation.

22.6 Register Description

Register base address: 0x40002000

Table 22-1: List of COMP Registers

Offset	Name	Description
0x118	COMP0_CFG	COMP0 control register
0x11C	COMP1_CFG	COMP1 control register
0x128	COMP0_POLL	COMP0 polling register
0x12C	COMP1_POLL	COMP1 polling register
0x138	COMP0_POLLMASK	COMP0 polling mask register
0x13C	COMP1_POLLMASK	COMP1 polling mask register
0x148	COMP0_LVSET	COMP0 filter configuration register
0x14C	COMP1_LVSET	COMP1 filter configuration register
0x158	COMP0_WINCFG	COMP0 window configuration register
0x15C	COMP1_WINCFG	COMP1 window configuration register
0x168	COMP0_INTSTAT	COMP0 interrupt status register

Offset	Name	Description
0x16C	COMP1_INTSTAT	COMP1 interrupt status register
0x178	COMP0_INTEN	COMP0 interrupt enable register
0x17C	COMP1_INTEN	COMP1 interrupt enable register

22.6.1 COMP0 Control Register COMP0_CFG (Offset: 118H)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	LOCK	RW	0x0	COMP0 output latch bit: 1: COMP0 maintains the current output value. 0: COMP0 outputs normally.
26:20	DA_VSET	RW	0x0	DAC output voltage setting: VOUT = ((DA_VSET + 1) / 128) * VREF
19:18	DA_VSSEL	RW	0x0	DAC reference voltage (VREF) selection: 00: VREF0 input, VREFLDO input 01: VREF1 input, PD3 input 10: VREF2 input, BUF 0.8 V/1.5 V input 11: Disabled
17	DA_EN	RW	0x0	DAC enable: 1: DAC enabled 0: DAC disabled
16:15	EDSET	RW	0x0	COMP0 trigger mode setting: 00: Level mode 01: Latch state on rising edge of COMP0 detection result 10: Latch state on falling edge of COMP0 detection result 11: Latch state on either rising or falling edge of COMP0 detection result
14:12	RSV	-	-	Reserved
11	COMNSEL	RW	0x0	Common-mode voltage setting bit for COMP0 P-terminals (P0, P1, P2): 1: Enable common mode voltage function. 0: Disable common mode voltage function.
10:9	HYS	RW	0x0	COMP0 hysteresis voltage setting bit: 00: 0 mV

Bit	Name	Attribute	Reset Value	Description
				01: 10 mV 10: 20 mV 11: 30 mV
8	POL	RW	0x0	COMP0 polarity selection: 1: COMP0 outputs 1 when the N-terminal voltage is high. 1: COMP0 outputs 1 when the P-terminal voltage is high.
7:5	INMSEL	RW	0x0	COMP0 negative input channel selection: 000: COMP0_N0 input, PC0 input 001: COMP0_N1 input, PB5 input 010: COMP0_N2 input, PB1 input 011: DAC input 100: Common-mode signal input
4:2	INPSEL	RW	0x0	COMP0 positive input channel selection: 000: COMP0_P0 input, PC1 input 001: COMP0_P1 input, PB7 input 010: COMP0_P2 input, PB6 input 011: COMP0_P3 input, PB4 input 100: COMP0_P4 input, PC2 input 101: COMP0_P5 input, PGA0_VOUT input 110: COMP0_P6 input, PGA1_VOUT input 111: COMP0_P7 input, PGA2_VOUT input
1	LMP	RW	0x0	COMP0 low-power mode control signal: 1: High speed 0: Low speed
0	EN	RW	0x0	COMP0 enable: 1: Enable COMP0 0: Disable COMP0

22.6.2 COMP1 Control Register COMP1_CFG (Offset: 11CH)

Bit	Name	Attribute	Reset Value	Description
31:28	RSV	-	-	Reserved
27	LOCK	RW	0x0	COMP1 output latch bit: 1: COMP1 maintains the current output value.

Bit	Name	Attribute	Reset Value	Description
				0: COMP1 outputs normally.
26:17	RSV	-	-	Reserved
16:15	EDSET	RW	0x0	COMP1 trigger mode setting: 00: Level mode 01: Latch state on rising edge of COMP1 detection result 10: Latch state on falling edge of COMP1 detection result 11: Latch state on either rising or falling edge of COMP1 detection result
14:12	RSV	-	-	Reserved
11	COMNSEL	RW	0x0	Common-mode voltage setting bit for COMP1 P-terminals (P0, P1, P2): 1: Enable common mode voltage function. 0: Disable common mode voltage function.
10:9	HYS	RW	0x0	COMP1 hysteresis voltage setting bit: 00: 0 mV 01: 10 mV 10: 20 mV 11: 30 mV
8	POL	RW	0x0	COMP1 polarity selection: 1: COMP1 outputs 1 when the N-terminal voltage is high. 1: COMP1 outputs 1 when the P-terminal voltage is high.
7:5	INMSEL	RW	0x0	COMP1 negative input channel selection: 000: COMP1_N0 input, PC0 input 001: COMP1_N1 input, PB5 input 010: COMP1_N2 input, PB3 input 011: DAC input 100: Common-mode signal input
4:2	INPSEL	RW	0x0	COMP1 positive input channel selection: 000: COMP1_P0 input, PB2 input 001: COMP1_P1 input, PB1 input 010: COMP1_P2 input, PB0 input 011: COMP1_P3 input, PB4 input 100: COMP1_P4 input, PB7 input

Bit	Name	Attribute	Reset Value	Description
				101: COMP1_P5 input, PGA0_VOUT input 110: COMP1_P6 input, PGA1_VOUT input 111: COMP1_P7 input, PGA2_VOUT input
1	LMP	RW	0x0	COMP1 low-power mode control signal: 1: High speed 0: Low speed
0	EN	RW	0x0	COMP1 enable: 1: Enable COMP1 0: Disable COMP1

22.6.3 COMP0 Polling Register COMP0_POLL (Offset: 128H)

Bit	Name	Attribute	Reset Value	Description
31:9	PERIOD	RW	0x0	Polling wait cycle. The time for each channel comparator is: 4 * t_{pclk} * COMP0_PERIOD
8:6	POUT	R	0x0	Polling channel output, reflecting the output status of the polling channel: Bit 6 corresponds to channel COMP0_P0 Bit 7 corresponds to channel COMP0_P1 Bit 8 corresponds to channel COMP0_P2 1: The comparator outputs 1. 0: The comparator outputs 0.
5:4	RSV	-	-	Reserved
3	POLMASKEN	RW	0x0	COMP0 polling mode mask function enable: 1: Enabled 0: Disabled
2	FIXN	RW	0x0	COMP0 polling channel inverted input fixed setting register: 1: The inverted input of polling channel is fixed. 0: The inverted input of polling channel is not fixed.
1	POLCH	RW	0x0	COMP0 polling channel setting: 1: Polling channels COMP0_P0/COMP0_P1/COMP0_P2

Bit	Name	Attribute	Reset Value	Description
				0: Polling channels COMP0_P0/COMP0_P1
0	POLEN	RW	0x0	COMP0 polling mode enable: 1: Enabled 0: Disabled

22.6.4 COMP1 Polling Register COMP1_POLL (Offset: 12CH)

Bit	Name	Attribute	Reset Value	Description
31:9	PERIOD	RW	0x0	Polling wait cycle. The time for each channel comparator is: 4 * t_{pclk} * COMP1_PERIOD
8:6	POUT	R	0x0	Polling channel output, reflecting the output status of the polling channel: Bit 6 corresponds to channel COMP1_P0 Bit 7 corresponds to channel COMP1_P1 Bit 8 corresponds to channel COMP1_P2 1: The comparator outputs 1. 0: The comparator outputs 0.
5:4	RSV	-	-	Reserved
3	POLMASKEN	RW	0x0	COMP1 polling mode mask function enable: 1: Enabled 0: Disabled
2	FIXN	RW	0x0	COMP1 polling channel inverted input fixed setting register: 1: The inverted input of polling channel is fixed. 0: The inverted input of polling channel is not fixed.
1	POLCH	RW	0x0	COMP1 polling channel setting: 1: Polling channels COMP1_P0/COMP1_P1/COMP1_P2 0: Polling channels COMP1_P0/COMP1_P1
0	POLEN	RW	0x0	COMP1 polling mode enable: 1: Enabled 0: Disabled

22.6.5 COMP0 Polling MASK Register COMP0_POLLMASK (Offset: 138H)

Bit	Name	Attribute	Reset Value	Description
31:22	RSV	-	-	Reserved
21:0	POLLMASK	RW	0x0	COMP0 polling mode channel switching mask time setting register The polling channel switching mask time is: t_{pclk} * COMP0_POLLMASK

22.6.6 COMP1 Polling MASK Register COMP1_POLLMASK (Offset: 13CH)

Bit	Name	Attribute	Reset Value	Description
31:22	RSV	-	-	Reserved
21:0	POLLMASK	RW	0x0	COMP1 polling mode channel switching mask time setting register The polling channel switching mask time is: t_{pclk} * COMP1_POLLMASK

22.6.7 COMP0 Filter Configuration Register COMP0_LVSET (Offset: 148h)

Bit	Name	Attribute	Reset Value	Description
31:24	RSV	-	-	Reserved
23:4	LVTSET	RW	0x0	COMP0 filter time setting register The COMP0 filter time is: t_{pclk} * COMP0_LVTSET
3	LV_CNT_RST	W	0x0	COMP0 software clear filter time counter Write 1 to clear this bit.
2:1	LVMSEL	RW	0x0	COMP0 filter mode selection: 00: Filter mode 0 01: Filter mode 1

Bit	Name	Attribute	Reset Value	Description
				10: Filter mode 2 11: Filter mode 3
0	LVEN	RW	0x0	COMP0 filter mode enable: 1: COMP0 filter function enabled 0: COMP0 filter function disabled

22.6.8 COMP1 Filter Configuration Register COMP1_LVSET (Offset: 14CH)

Bit	Name	Attribute	Reset Value	Description
31:24	RSV	-	-	Reserved
23:4	LVTSET	RW	0x0	COMP1 filter time setting register The COMP1 filter time is: t_{pclk} * COMP1_LVTSET
3	LV_CNT_RST	W	0x0	COMP1 software clear filter time counter Write 1 to clear the bit.
2:1	LVMSEL	RW	0x0	COMP1 filter mode selection: 00: Filter mode 0 01: Filter mode 1 10: Filter mode 2 11: Filter mode 3
0	LVEN	RW	0x0	COMP1 filter mode enable: 1: COMP1 filter function enabled 0: COMP1 filter function disabled

22.6.9 COMP0 WINDOW Configuration Register COMP0_WINCFG (Offset: 158H)

Bit	Name	Attribute	Reset Value	Description
31:24	WINSET	RW	0x0	Window time setting bit The window time is: 2 * t_{pclk} * (WINSET + 1)
23:22	WINEDS	RW	0x0	Window PWM edge setting bit: 00: PWM does not participate in window function.

Bit	Name	Attribute	Reset Value	Description
				01: PWM falling edge participates in window function. 10: PWM rising edge participates in window function. 11: PWM both edges participate in window function.
21	LPTC2S	RW	0x0	LPTIM_CH2 PWM window function setting bit: 1: LPTIMER_CH2 participates in window function. 0: LPTIMER_CH2 does not participate in window function.
20	LPTC1S	RW	0x0	LPTIMER_CH1 PWM window function setting bit: 1: LPTIMER_CH1 participates in window function. 0: LPTIMER_CH1 does not participate in window function.
19	ATOC4S	RW	0x0	ATIMER OC4REF PWM window function setting bit: 1: ATIMER OC4REF participates in window function. 0: ATIMER OC4REF does not participate in window function.
18	ATOC3NS	RW	0x0	ATIMER_CH3N PWM window function setting bit: 1: ATIMER_CH3N participates in window function. 0: ATIMER_CH3N does not participate in window function.
17	ATOC3S	RW	0x0	ATIMER OC3REF PWM window function setting bit: 1: ATIMER OC3REF participates in window function. 0: ATIMER OC3REF does not participate in window function.
16	ATOC2NS	RW	0x0	ATIMER_CH2N PWM window function setting bit: 1: ATIMER_CH2N participates in window function. 0: ATIMER_CH2N does not participate in window function.
15	ATOC2S	RW	0x0	ATIMER OC2REF PWM window function setting bit: 1: ATIMER OC2REF participates in window function. 0: ATIMER OC2REF does not participate in window function.
14	ATOC1NS	RW	0x0	ATIMER_CH1N PWM window function setting bit: 1: ATIMER_CH1N participates in window function. 0: ATIMER_CH1N does not participate in window function.
13	ATOC1S	RW	0x0	ATIMER OC1REF PWM window function setting bit:

Bit	Name	Attribute	Reset Value	Description
				1: ATIMER OC1REF participates in window function. 0: ATIMER OC1REF does not participate in window function.
12	GT2OC4S	RW	0x0	GTIMER2 OC4REF PWM window function setting bit: 1: GTIMER2 OC4REF participates in window function. 0: GTIMER2 OC4REF does not participate in window function.
11	GT2OC3S	RW	0x0	GTIMER2 OC3REF PWM window function setting bit: 1: GTIMER2 OC3REF participates in window function. 0: GTIMER2 OC3REF does not participate in window function.
10	GT2OC2S	RW	0x0	GTIMER2 OC2REF PWM window function setting bit: 1: GTIMER2 OC2REF participates in window function. 0: GTIMER2 OC2REF does not participate in window function.
9	GT2OC1S	RW	0x0	GTIMER2 OC1REF PWM window function setting bit: 1: GTIMER2 OC1REF participates in window function. 0: GTIMER2 OC1REF does not participate in window function.
8	GT1OC4S	RW	0x0	GTIMER1 OC4REF window function setting bit: 1: GTIMER1 OC4REF participates in window function. 0: GTIMER1 OC4REF does not participate in window function.
7	GT1OC3S	RW	0x0	GTIMER1 OC3REF window function setting bit: 1: GTIMER1 OC3REF participates in window function. 0: GTIMER1 OC3REF does not participate in window function.
6	GT1OC2S	RW	0x0	GTIMER1 OC2REF window function setting bit: 1: GTIMER1 OC2REF participates in window function. 0: GTIMER1 OC2REF does not participate in window function.
5	GT1OC1S	RW	0x0	GTIMER1 OC1REF window function setting bit: 1: GTIMER1 OC1REF participates in window function. 0: GTIMER1 OC1REF does not participate in window function.
4	GT0OC4S	RW	0x0	GTIMER0 OC4REF window function setting bit:

Bit	Name	Attribute	Reset Value	Description
				1: GTIMER0 OC4REF participates in window function. 0: GTIMER0 OC4REF does not participate in window function.
3	GT0OC3S	RW	0x0	GTIMER0 OC3REF window function setting bit: 1: GTIMER0 OC3REF participates in window function. 0: GTIMER0 OC3REF does not participate in window function.
2	GT0OC2S	RW	0x0	GTIMER0 OC2REF window function setting bit: 1: GTIMER0 OC2REF participates in window function. 0: GTIMER0 OC2REF does not participate in window function.
1	GT0OC1S	RW	0x0	GTIMER0 OC1REF window function setting bit: 1: GTIMER0 OC1REF participates in window function. 0: GTIMER0 OC1REF does not participate in window function.
0	WINEN	RW	0x0	Comparator 0 window function enable: 1: Enable window function. 0: Disable window function.

22.6.10 COMP1 WINDOW Configuration Register COMP1_WINCFG (Offset: 15CH)

Bit	Name	Attribute	Reset Value	Description
31:24	WINSET	RW	0x0	Window time setting bit The window time is: 2 * t_{pclk} * (WINSET + 1)
23:22	WINEDS	RW	0x0	Window PWM edge setting bit: 00: PWM does not participate in window function. 01: PWM falling edge participates in window function. 10: PWM rising edge participates in window function. 11: PWM both edges participate in window function.

Bit	Name	Attribute	Reset Value	Description
21	LPTC2S	RW	0x0	LPTIMER_CH2 PWM window function setting bit: 1: LPTIMER_CH2 participates in window function. 0: LPTIMER_CH2 does not participate in window function.
20	LPTC1S	RW	0x0	LPTIMER_CH1 PWM window function setting bit: 1: LPTIMER_CH1 participates in window function. 0: LPTIMER_CH1 does not participate in window function.
19	ATOC4S	RW	0x0	ATIMER OC4REF PWM window function setting bit: 1: ATIMER OC4REF participates in window function. 0: ATIMER OC4REF does not participate in window function.
18	ATC3NS	RW	0x0	ATIMER_CH3N PWM window function setting bit: 1: ATIMER_CH3N participates in window function. 0: ATIMER_CH3N does not participate in window function.
17	ATOC3S	RW	0x0	ATIMER OC3REF PWM window function setting bit: 1: ATIMER OC3REF participates in window function. 0: ATIMER OC3REF does not participate in window function.
16	ATC2NS	RW	0x0	ATIMER_CH2N PWM window function setting bit: 1: ATIMER_CH2N participates in window function. 0: ATIMER_CH2N does not participate in window function.
15	ATOC2S	RW	0x0	ATIMER OC2REF PWM window function setting bit: 1: ATIMER OC2REF participates in window function. 0: ATIMER OC2REF does not participate in window function.
14	ATOC1NS	RW	0x0	ATIMER_CH1N PWM window function setting bit: 1: ATIMER_CH1N participates in window function. 0: ATIMER_CH1N does not participate in window function.

Bit	Name	Attribute	Reset Value	Description
13	ATOC1S	RW	0x0	ATIMER OC1REF PWM window function setting bit: 1: ATIMER OC1REF participates in window function. 0: ATIMER OC1REF does not participate in window function.
12	GT2OC4S	RW	0x0	GTIMER2 OC4REF PWM window function setting bit: 1: GTIMER2 OC4REF participates in window function. 0: GTIMER2 OC4REF does not participate in window function.
11	GT2OC3S	RW	0x0	GTIMER2 OC3REF PWM window function setting bit: 1: GTIMER2 OC3REF participates in window function. 0: GTIMER2 OC3REF does not participate in window function.
10	GT2OC2S	RW	0x0	GTIMER2 OC2REF PWM window function setting bit: 1: GTIMER2 OC2REF participates in window function. 0: GTIMER2 OC2REF does not participate in window function.
9	GT2OC1S	RW	0x0	GTIMER2 OC1REF PWM window function setting bit: 1: GTIMER2 OC1REF participates in window function. 0: GTIMER2 OC1REF does not participate in window function.
8	GT1OC4S	RW	0x0	GTIMER1 OC4REF window function setting bit: 1: GTIMER1 OC4REF participates in window function. 0: GTIMER1 OC4REF does not participate in window function.

Bit	Name	Attribute	Reset Value	Description
7	GT1OC3S	RW	0x0	GTIMER1 OC3REF window function setting bit: 1: GTIMER1 OC3REF participates in window function. 0: GTIMER1 OC3REF does not participate in window function.
6	GT1OC2S	RW	0x0	GTIMER1 OC2REF window function setting bit: 1: GTIMER1 OC2REF participates in window function. 0: GTIMER1 OC2REF does not participate in window function.
5	GT1OC1S	RW	0x0	GTIMER1 OC1REF window function setting bit: 1: GTIMER1 OC1REF participates in window function. 0: GTIMER1 OC1REF does not participate in window function.
4	GT0OC4S	RW	0x0	GTIMER0 OC4REF window function setting bit: 1: GTIMER0 OC4REF participates in window function. 0: GTIMER0 OC4REF does not participate in window function.
3	GT0OC3S	RW	0x0	GTIMER0 OC3REF window function setting bit: 1: GTIMER0 OC3REF participates in window function. 0: GTIMER0 OC3REF does not participate in window function.
2	GT0OC2S	RW	0x0	GTIMER0 OC2REF window function setting bit: 1: GTIMER0 OC2REF participates in window function. 0: GTIMER0 OC2REF does not participate in window function.
1	GT0OC1S	RW	0x0	GTIMER0 OC1REF window function setting bit: 1: GTIMER0 OC1REF participates in window function. 0: GTIMER0 OC1REF does not participate in window function.

Bit	Name	Attribute	Reset Value	Description
0	WINEN	RW	0x0	Comparator 0 window function enable: 1: Enable window function. 0: Disable window function.

22.6.11 COMP0 Status Register COMP0_INTSTAT (Offset: 168H)

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9	ARST	R	0x0	COMP0 analog part real-time status register: 1: The current comparison result of the COMP0 analog part is 1. 0: The current comparison result of the COMP0 analog part is 0.
8	RST	R	0x0	COMP0 real-time status register: 1: The current comparison result of COMP0 is 1. 0: The current comparison result of COMP0 is 0.
7	RSV	-	-	Reserved
6	PP2INST	W1C	0x0	COMP0_P2 channel interrupt status register in polling mode: 1: A comparison result of 1 is generated. 1: No comparison result of 1 is generated. Write 1 to clear this bit.
5	PP1INST	W1C	0x0	COMP0_P1 channel interrupt status register in polling mode: 1: A comparison result of 1 is generated. 1: No comparison result of 1 is generated. Write 1 to clear this bit.
4	PP0INST	W1C	0x0	COMP0_P0 channel interrupt status register in polling mode: 1: A comparison result of 1 is generated. 1: No comparison result of 1 is generated. Write 1 to clear this bit.
3:1	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
0	INTST	W1C	0x0	COMP0 interrupt status register: 1: COMP0 generates a comparison result of 1. 0: COMP0 does not generate a comparison result of 1. Write 1 to clear the bit.

22.6.12 COMP1 Status Register COMP1_INTSTAT (Offset: 16CH)

Bit	Name	Attribute	Reset Value	Description
31:10	RSV	-	-	Reserved
9	ARST	R	0x0	COMP1 analog part real-time status register: 1: The current comparison result of the COMP1 analog part is 1. 0: The current comparison result of the COMP1 analog part is 0.
8	RST	R	0x0	COMP1 real-time status register: 1: The current comparison result of COMP1 is 1. 0: The current comparison result of COMP1 is 0.
7	RSV	-	-	Reserved
6	PP2INST	W1C	0x0	COMP1_P2 channel interrupt status register in polling mode: 1: A comparison result of 1 is generated. 1: No comparison result of 1 is generated. Write 1 to clear the bit.
5	PP1INST	W1C	0x0	COMP1_P1 channel interrupt status register in polling mode: 1: A comparison result of 1 is generated. 1: No comparison result of 1 is generated. Write 1 to clear the bit.
4	PP0INST	W1C	0x0	COMP1_P0 channel interrupt status register in polling mode: 1: A comparison result of 1 is generated. 1: No comparison result of 1 is generated. Write 1 to clear the bit.
3:1	RSV	-	-	Reserved

Bit	Name	Attribute	Reset Value	Description
0	INTST	W1C	0x0	COMP1 interrupt status register: 1: COMP1 generates a comparison result of 1. 0: COMP1 does not generate a comparison result of 1. Write 1 to clear the bit.

22.6.13 COMP0 Interrupt Enable Register COMP0_INTEN (Offset: 178H)

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	PP2INTEN	RW	0x0	COMP0_PP2INST interrupt enable register: 1: COMP0_PP2INST interrupt enabled 0: COMP0_PP2INST interrupt disabled
5	PP1INTEN	RW	0x0	COMP0_PP1INST interrupt enable register: 1: COMP0_PP1INST interrupt enabled 0: COMP0_PP1INST interrupt disabled
4	PP0INTEN	RW	0x0	COMP0_PP0INST interrupt enable register: 1: COMP0_PP0INST interrupt enabled 0: COMP0_PP0INST interrupt disabled
3:1	RSV	-	-	Reserved
0	INTEN	RW	0x0	COMP0_INTST interrupt enable register: 1: COMP0_INTST interrupt enabled 0: COMP0_INTST interrupt disabled

22.6.14 COMP1 Interrupt Enable Register COMP1_INTEN (Offset: 17CH)

Bit	Name	Attribute	Reset Value	Description
31:7	RSV	-	-	Reserved
6	PP2INTEN	RW	0x0	COMP1_PP2INST interrupt enable register: 1: COMP1_PP2INST interrupt enabled 0: COMP1_PP2INST interrupt disabled

Bit	Name	Attribute	Reset Value	Description
5	PP1INTEN	RW	0x0	COMP1_PP1INST interrupt enable register: 1: COMP1_PP1INST interrupt enabled 0: COMP1_PP1INST interrupt disabled
4	PP0INTEN	RW	0x0	COMP1_PP0INST interrupt enable register: 1: COMP1_PP0INST interrupt enabled 0: COMP1_PP0INST interrupt disabled
3:1	RSV	-	-	Reserved
0	INTEN	RW	0x0	COMP1_INTST interrupt enable register: 1: COMP1_INTST interrupt enabled 0: COMP1_INTST interrupt disabled

22.7 Operation Procedure

Basic configuration for comparators:

1. Select the positive and negative input channels and trigger mode in the COMPx_CFG register.
2. In the COMPx_POLL register, it is possible to choose whether to enable the polling mode and configure the polling wait cycle.
3. Set the polling channel switching time in the COMPx_POLLMASK register.
4. In the COMPx_LVSET register, select the filtering function, including filtering mode and filtering time.
5. In the COMPx_WINCFG register, select the window trigger bit, window time, and window enable.
6. Enable interrupts in the WWDT_IE register.
7. Enable COMPx_EN and observe the interrupt comparison results.

23 DIV

23.1 Overview

The implementation goal of DIV is to support division operations with divisors not exceeding 32 bits. The divisor can be up to 32 bits, while the dividend can be of any bit length. It is primarily used in applications where the divisor is less than 32 bits.

23.2 Main Features

- Signed integer operations
- Supporting divisions with divisors not exceeding 32 bits
- Division-by-zero warning function
- Output the calculation result within 8 AHB clock cycles

23.3 Register Description

DIV module base address: 0x4002_1000

Table 23-1: List of DIV Registers

Offset	Name	Description
0x00	DIV_DIVIDEND	Dividend register
0x04	DIV_DIVISOR	Divisor register
0x08	DIV_REMAIN	Remainder register
0x0C	DIV_QUOTIENT	Quotient register
0x10	DIV_STATUS	Status register

23.3.1 Dividend Register DIV_DIVIDEND (Offset: 00H)

Bit	Name	Attribute	Reset Value	Description
31:0	DIVIDEND	RW	0x0	Dividend

23.3.2 Divisor Register DIV_DIVISOR (Offset: 04H)

Bit	Name	Attribute	Reset Value	Description
31:0	DIVISOR	RW	0x1	Divisor

23.3.3 Remainder Register DIV_REMAIN (Offset: 08H)

Bit	Name	Attribute	Reset Value	Description
31:0	REMAIN	R	0x0	Remainder

23.3.4 Quotient Register DIV_QUITIENT (Offset: 0CH)

Bit	Name	Attribute	Reset Value	Description
31:0	QUOTIENT	R	0x0	Quotient

23.3.5 Status Register DIV_STATUS (Offset: 10H)

Bit	Name	Attribute	Reset Value	Description
31:2	RSV	-	-	Reserved
1	DB0	R	0x0	Divisor flag: 1: Divisor is 0. 0: Divisor is not 0.
0	DONE	R	0x1	Operation completion flag: 1: Operation is completed. 0: Operation is not completed.

23.4 Operation Procedure

1. Write the divisor into the divisor register.
2. Write the dividend into the dividend register.
3. Read the quotient from the quotient register every 8 algorithm clock cycles or by querying the completion status bit. If the dividend has not been fully input, return to step 2; if it is fully input, proceed to step 4.
4. Read the remainder from the remainder register, and the operation ends.

24 SQRT

24.1 Overview

The SQRT module is a square root operation module.

24.2 Main Features

- AHB bus interface, with results output within 4 AHB bus clock cycles

24.3 Register Description

SQRT register base address: 0x4002_1400

Table 24-1: List of SQRT Registers

Offset	Name	Description
0x00	SQRT_VALUE	SQRT radicand register
0x04	SQRT_RESULT	SQRT square root operation result register
0x08	SQRT_SR	SQRT status register

24.3.1 SQRT Radicand Register (SQRT_VALUE) (Offset: 00H)

Bit	Name	Attribute	Reset Value	Description
31:0	VALUE	RW	0x0	SQRT radicand register: Writing to this register starts the square root operation immediately with the written data. The result is output within 4 AHB clock cycles.

24.3.2 SQRT Square Root Operation Result Register (SQRT_RESULT) (Offset: 04H)

Bit	Name	Attribute	Reset Value	Description
31:0	RESULT	R	0x0	This register stores the operation result after the square root operation is completed.

24.3.3 SQRT Status Register (SQRT_SR) (Offset: 08H)

Bit	Name	Attribute	Reset Value	Description
31:1	RSV	-	-	Reserved
0	BUSY	R	0x0	Square root operation status flag: 1: Square root operation is in progress, not completed. 0: Square root operation is completed, hardware is idle.

24.4 Operation Procedure

1. Write the radicand value to SQRT_VALUE.
2. Perform a software delay or poll the SQRT_SR register and wait for the value to be 0.
3. Read SQRT_RESULT to obtain the square root result.

25 SysTick

25.1 Overview

To support multitasking, an OS needs to perform context switching periodically, which requires hardware resources such as timers to interrupt program execution. When a timer interrupt occurs, the processor performs OS task scheduling in the exception handling routine and also carries out maintenance work for the OS. The Cortex-M0 + processor features a simple timer called SysTick, which is used to generate periodic interrupt requests.

SysTick is a 24-bit timer that counts downwards. When the timer counts down to 0, it reloads a programmable value and simultaneously generates a SysTick exception (exception number 15). This exception event triggers the execution of SysTick exception handler, which is part of the OS.

For systems that do not require an OS, the SysTick timer can also be used for other purposes, such as timing, counting, or providing an interrupt source for tasks that need to be executed periodically. The generation of SysTick exception is controllable. If the exception is disabled, the SysTick timer can still be used in a polling way, such as checking the current count value or polling the overflow flag.

25.2 Register Description

SysTick register base address: 0xE000_E010

Table 25-1: List of SysTick Registers

Offset	Name	Description
0x00	SysTick_CTRL	SysTick control and status register
0x04	SysTick_LOAD	SysTick reload value register
0x08	SysTick_VAL	SysTick current value register

25.2.1 Control and Status Register SysTick_CTRL (Offset: 00H)

Bit	Name	Attribute	Reset Value	Description
31:17	RSV	-	-	Reserved
16	COUNTFLAG	R	0x0	SysTick timer underflow flag: 1: SysTick timer has underflowed. 0: SysTick timer has not underflowed. Reading this register will clear the COUNTFLAG bit.
15:2	RSV	-	-	Reserved
1	TICKINT	RW	0x0	SysTick interrupt enable: 1: Enable interrupt 0: Disable interrupt
0	ENABLE	RW	0x0	SysTick timer enable: 1: Enable SysTick 0: Disable SysTick

25.2.2 Reload Value Register SysTick_LOAD (Offset: 04H)

Bit	Name	Attribute	Reset Value	Description
31:24	RSV	-	-	Reserved
23:0	RELOAD	RW	0xFFFFFF	SysTick timer reload value

25.2.3 Current Value Register SysTick_VAL (Offset: 08H)

Bit	Name	Attribute	Reset Value	Description
31:24	RSV	-	-	Reserved
23:0	CURRENT	RW	0xFFFFFF	Reading this register retrieves the current count value of the SysTick timer. Writing any value to this register clears both the register and the COUNTFLAG.

25.3 Operation Procedure

Since both the reload value and current value of SysTick timer are undefined at reset, to prevent abnormal results, the configuration of SysTick must follow a specific procedure:

1. Set SysTick_CTRL[0] to 0 to disable SysTick.
2. Configure SysTick_LOAD to select the overflow period of SysTick.
3. Write any value to SysTick_VAL to clear SysTick_VAL and SysTick_CTRL.
4. Check COUNTFLAG.
5. Set SysTick_CTRL[1] to 1 to enable the SysTick interrupt.
6. Set SysTick_CTRL[0] to 1 to enable SysTick.
7. Either poll and wait for the timer overflow flag, then disable and clear the counter, or read SysTick_CTRL in the interrupt service routine to clear the overflow flag.

26 Debug Support (DBG)

The Cortex®-M0 core with an integrated hardware debug module supports instruction breakpoints (stopping on instruction fetch) and data breakpoints (stopping on data access). When the core is halted, users can inspect the internal state of the core and the external state of the system. Upon completion of the query operations, the core and peripherals can be resumed to continue executing the corresponding program. The hardware debugging module of the chip core is available when connected to a debugger (unless disabled).

Serial interface (two-wire SWD) for debugging is available.

27 Revision History

Date	Version	Description
Nov-18-2024	V1.0	Initial release.
Sep-10-2025	V1.0.1	1. Modified the OTP_MAIN_EN[15] control address range. 2. Updated the description of some registers. 3. Updated the maximum system frequency to 60 MHz.